UNIVERSIDADE DE BRASÍLIA - UnB

INSTITUTO DE CIÊNCIA POLÍTICA - IPOL

PROGRAMA DE PÓS-GRADUAÇÃO EM DEMOCRACIA E SOCIEDADE

Comunidade Noosfero: Um estudo de caso sobre o papel da agência criativa na preservação da unidade de uma comunidade FLOSS

Ricardo Augusto Poppi Martins

Brasília

Abril de 2019

UNIVERSIDADE DE BRASÍLIA - UnB
INSTITUTO DE CIÊNCIA POLÍTICA - IPOL
PROGRAMA DE PÓS-GRADUAÇÃO EM DEMOCRACIA E SOCIEDADE

Comunidade Noosfero: Um estudo de caso sobre o papel da agência criativa na preservação da unidade de uma comunidade FLOSS

Dissertação apresentada ao Instituto de Ciência Política da Universidade de Brasília (UnB) como requisito à obtenção do título de mestre em Ciência Política, sob a orientação Professora Doutora Marisa von Bülow.

Ricardo Augusto Poppi Martins

Brasília
Abril de 2019

UNIVERSIDADE DE BRASÍLIA - UnB
INSTITUTO DE CIÊNCIA POLÍTICA - IPOL
PROGRAMA DE PÓS-GRADUAÇÃO EM DEMOCRACIA E SOCIEDADE

Comunidade Noosfero: Um estudo de caso sobre o papel da agência criativa na preservação da unidade de uma comunidade FLOSS

Banca examinadora:

Profa. Dra. Marisa von Bülow, orientadora
Profa. Dra Carla Rocha, membro externo
Profa. Dra. Rebecca Abers, membro interno
Profa. Dra. Debora Rezende de Almeida, suplente

Ricardo Augusto Poppi Martins

Brasília
Abril de 2019

"With enough of us, around the world, we'll not just send a strong message opposing the privatization of knowledge — we'll make it a thing of the past. Will you join us?"

Aaron Swartz July 2008, Eremo, Italy

para Aaron Swartz (em memória), o menino da internet

AGRADECIMENTOS

Essa pesquisa é fruto de um trabalho coletivo construído em cima da obra e dos ensinamentos de muites que se dedicaram e se dedicam a fazer ciência. Sou o único responsável por todos os erros e imprecisões que um trabalho como esse acaba carregando, mas o valor que essa pesquisa possa vir a ter certamente se deve também aqueles que me acompanharam de alguma forma nessa trajetória. Não seria possível agradecer a todes que de alguma forma contribuíram para que pudesse realizar esse trabalho então vou me concentrar naqueles com quem eu tive o privilégio de compartilhar discussões, inquietações e alegrias durante os mais de dois anos em que estive especialmente dedicado a essa pesquisa. Agradeço à minha esposa Gardênia pelo apoio nos diversos momentos de angústia com a pesquisa e por me motivar a frequentar a biblioteca central da UnB, inclusive algumas vezes com a sua companhia, onde eu aprendi a receber os efeitos benéficos da psicologia social proporcionada por dezenas de pessoas dedicadas aos seus estudos, num ambiente ímpar para concentração e produção intelectual. Com isso agradeço também a todos os funcionários da Universidade de Brasília por manter essa e outras estruturas funcionando, tão fundamentais para a boa formação de ensino superior e para a produção científica do Brasil.

Agradeço também aos membros do Resocie, o grupo de pesquisa do IPOL/UnB, que desde o início me acolheu (alô Ale Gomes!) proporcionando acesso a discussões de altíssimo nível e aprendizados fundamentais para quem está praticando o ofício de pesquisador. Deles, especialmente a minha orientadora Professora Marisa von Bülow pela generosa dedicação a me orientar com dicas valiosas durante toda as etapas desse processo e pela paciência com as minhas ausências, inevitáveis para quem combina o mestrado com uma série de outras atividades profissionais e políticas. Do grupo, agradeço também à Professora Rebecca Abers pelas ricas discussões sobre criatividade política e pelas contribuições de alto nível no momento de minha qualificação.

Agradeço também aos Professores Hilmer, Paulo, Carla e Fábio do Laboratório Lappis da FGA/UnB, talentosos professores de engenharia de software que me proporcionaram contato permanente com o mundo do desenvolvimento de software e das comunidades FLOSS, assim como com os princípios do movimento Software Livre. Estendo os agradecimentos aos meus outros colegas de laboratório Joenio Costa e Valéssio Brito (também membros da comunidade Noosfero) pelo companheirismo e paciência com nossas

discussões sobre o mundo da colaboração aberta. Também agradeço ao Professor Dalton Martins da FCI/UnB pelos insights apontados na minha banca de qualificação, especialmente voltados a literatura sobre o Comum.

Agradeço também à Daniela Feitosa, Leandro Nunes e Caiosba, membros da comunidade Noosfero, com quem tive o privilégio de trabalhar nos projetos que envolviam as plataformas de participação social do governo federal e com quem aprendi bastante. Em nome deles, agradeço aos mais de 15 membros da comunidade que, com bastante generosidade, dedicaram uma parte grande de seu tempo em me relatar tudo que puderam sobre o Noosfero durante as entrevistas para essa pesquisa. A receptividade dos membros da comunidade para as minhas investidas foi enorme o que além de facilitar o trabalho da pesquisa, demonstrou o perfil aberto e solícito daqueles que engajam nesse tipo de empreendimento livre e solidário.

Agradeço também ao meu colega de Instituto Cidade Democrática e Empurrando Juntas, Henrique Parra Parra, pela parceria nas discussões sobre o tema de tecnologia e Estado, pelas dicas de bibliografia que passaram a fazer parte da dissertação e pela paciência com minhas ausências na reta final da sistematização e escrita dessa pesquisa.

RESUMO

A internet trouxe o fenômeno da colaboração aberta e da codificação social onde milhares de pessoas trabalham na construção colaborativa de softwares que estão sendo cada vez mais adotados por governos do mundo todo. Essa pesquisa faz um estudo de caso do Noosfero, uma comunidade FLOSS Brasileira, buscando identificar como a agência criativa dos indivíduos, através da utilização de certos comportamentos e táticas, buscou influenciar as decisões sobre o software mantido pela comunidade. Analisando dados do repositório de código da comunidade e a realização de 15 entrevistas semi-estruturadas com membros da comunidade, a pesquisa identificou que a agência criativa dos membros foi responsável pela manutenção da unidade da comunidade ao longo dos seus mais de 11 anos de existência. Além disso a pesquisa busca oferecer um guia para membros de organizações que desejam ou precisam se relacionar com comunidades FLOSS.

Palavras chave: FLOSS, colaboração aberta, codificação social, autoridade lateral, autoridade prática, criatividade política

ABSTRACT

The internet has brought the phenomenon of open collaboration and social coding where thousands of people work in the collaborative construction of software that is increasingly being adopted by governments around the world. This research is a case study of Noosfero, a Brazilian FLOSS community, seeking to identify how the creative agency of individuals, through the use of certain behaviors and tactics, sought to influence decisions about software maintained by the community. Analyzing data from the community code repository and performing 15 semi-structured interviews with community members, the research identified that the creative agency of the members was responsible for maintaining the unity of the community throughout its more than 11 years of existence. In addition, the research seeks to provide a guide for members of organizations that want or need to create relationships with FLOSS communities.

Keywords: FLOSS, open collaboration, social coding, lateral authority, practical authority, political creativity

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. Introdução

A onda avassaladora da internet e da web trouxe junto a promessa de uma sociedade mais inclusiva, democrática e plural, produzindo um certo encantamento tanto em pesquisadores que discutem os impactos dessas tecnologias na sociedade, quanto em movimentos ligados aos princípios da cultura hacker, cultura digital e software livre. Alguns autores defendem que as novas possibilidades de comunicação e organização trazidas pela internet podem promover a inclusão e a participação maior das pessoas em diversas áreas antes restritas aos especialistas (SHIRKY 2012, BENKLER, 2006). Fazem parte desse fenômeno as transformações na forma com que a sociedade humana se relaciona e produz conhecimento aberto e os arranjos de propriedade intelectual com incentivos voltados à distribuição e produção compartilhada (CASTELLS, 2003; BENKLER, 2006).

A partir de uma abordagem mais específica das práticas potencialmente transformadoras no contexto do setor público, surgem pesquisas sobre o tema da "Colaboração Aberta" ou "Inovação Colaborativa" (SØRENSEN & TORFING, 2011 ; FORTE & LAMPE, 2013 ; MERGEL, 2015). Segundo Sørensen & Torfing (2011), esse fenômeno se caracteriza como uma "fonte de inovação pública fornecida por formas de colaboração baseadas em rede, que podem compensar as deficiências das hierarquias e dos mercados" (página 845, tradução nossa). Forte & Lampe (2013) definem colaboração aberta como

> (a) um ambiente online que suporta a produção coletiva de um artefato (b) através de uma plataforma de colaboração tecnologicamente mediada (c) que apresenta baixas barreiras de entrada e saída e (d) suporta a emergência de estruturas sociais persistentes, mas maleáveis (FORTE & LAMPE, 2013, página 536, tradução nossa).

Os autores ainda apontam que há uma estratégia inscrita nesse fenômeno, baseada em "produzir inovação usando a Internet para convidar co-criadores desconhecidos de todo o mundo para ajudar na solução de um problema específico" (SØRENSEN & TORFING, 2011, página 853, tradução nossa).

Mergel (2015) define um tipo específico desse fenômeno mais amplo de colaboração aberta como "*social coding*" ou codificação social, que é quando desenvolvedores e usuários colaboram na construção dos códigos das aplicações digitais. Esse tipo de colaboração aberta

se manifesta na construção dos softwares de código aberto, hoje responsáveis por boa parte das aplicações que viabilizam a Internet. Essa prática de construção compartilhada, participativa e distribuída (WEBER, 2004; MEIRELES, 2015; GERMANI, 2016), colocada como sinônimo de abertura para participação dos cidadãos e alto grau de transparência, inspira recomendações para que seja adotada por governos e instituições Estatais em geral (MEIRELES, 2015; GERMANI, 2016). A partir da análise do uso da maior plataforma de colaboração em código no mundo, o Github[1], Mergel (2015) sugere que a utilização de ambientes de codificação social representam formas inovadoras de colaboração no contexto governamental, a partir da reutilização de código entre os órgãos Estatais (página 471, tradução nossa). Segundo a autora, ainda que os resultados sejam pouco conhecidos até o momento, esse fenômeno tem gerado um aumento de transparência nas ações governamentais (idem, página 471, tradução nossa).

Numa linha parecida, Criado (2016) acredita que as plataformas de colaboração aberta (a exemplo do Github) produzem inovação colaborativa dentro das Instituições Estatais. Segundo o autor, essas plataformas são utilizadas tanto para colaboração entre atores de uma mesma organização, como por membros de organizações externas (página 264). Falando especificamente a partir de seus achados nas análises sobre o Github, o pesquisador acredita que o uso da plataforma pode gerar valiosas dinâmicas colaborativas tanto para o setor público quanto para a cidadania:

> "O caso do GitHub demonstra o potencial para a colaboração aberta entre servidores públicos dentro de uma plataforma aberta na qual podem interagir com outros profissionais do setor para desenvolver projetos conjuntamente, tanto de software quanto de outro tipo" (CRIADO, 2016, página 264, tradução nossa).

Para que seja viável o trabalho de criação de softwares nas plataformas de codificação social, é fundamental que todos os desenvolvedores (e potencialmente qualquer pessoa) possam ter acesso ao código fonte dos programas. O código fonte de um programa são as instruções que fazem aquele programa funcionar. É no código fonte que está todo o conhecimento contido no software. É através dele que outros desenvolvedores podem estudar,

[1] O Github é a maior plataforma de colaboração aberta em código no mundo. Mantida por uma empresa norte-americana, a plataforma permite que os desenvolvedores e organizações publiquem seus códigos e colaborem nos códigos publicados pelos outros desenvolvedores. Embora seja uma plataforma de compartilhamento e colaboração em código, ela possui funções típicas de mídias sociais, como curtidas e menções, facilitando a gestão colaborativa das propostas de melhorias e de tarefas. Link de acesso: https://github.com/

aprimorar ou customizar o software para seus usos específicos. Para que o código fonte do software esteja disponível de forma segura e duradoura a todos os interessados, os desenvolvedores utilizam o instrumento jurídico das licenças. Uma licença é uma espécie de contrato onde os criadores do software explicitam o que as pessoas podem fazer com o código, considerado como propriedade intelectual.

Porém, tanto as licenças de software livre quanto as de "open source" garantem o acesso de qualquer pessoa ao código. Os conceitos de software livre e "*open source*" (código aberto), ainda que bastante parecidos, possuem algumas distinções importantes. O software livre é um qualificador usado para denominar os códigos produzidos e licenciados por comunidades de desenvolvedores e usuários que se posicionam em torno da liberdade do acesso e apropriação do conhecimento contido nos softwares. Para isso, utilizam um mecanismo jurídico conhecido como licenças virais, ou seja, contratos públicos que obrigam todos aqueles que aceitarem trabalhar com aquele código a publicar suas contribuições de forma aberta e pública garantindo, assim, as 4 liberdades básicas do software livre (executar, estudar, distribuir e modificar)[2]. Essas licenças utilizam a prerrogativa do direito de propriedade para obrigar todas as pessoas envolvidas nas comunidades a manterem esse conhecimento aberto. O movimento do software livre, com seu recorte programático voltado para a liberdade do conhecimento, e a sua principal licença viral, a GPL[3] constituem os principais elementos do conceito de software livre.

Por outro lado, o conceito de "*open source*" ou "código aberto", que surge a partir de uma dissidência do movimento Software Livre, é adotado por desenvolvedores e comunidades que priorizam os aspectos comerciais do software de código aberto, acreditando que as pessoas e organizações não devem ser obrigadas a manter seus códigos abertos, mas podem estar livres para fazer isso por iniciativa própria. Dessa forma, as pessoas que advogam pelo software de código aberto e suas licenças derivadas não têm como característica comum uma ideologia voltada ao conhecimento aberto, mas sim um outro tipo de ideologia que se organiza em torno da ideia de que o software de código aberto é melhor

---

[2] (1) A liberdade de executar o programa, para qualquer propósito; (2) A liberdade de estudar o programa, e adaptá-lo para as suas necessidades; (3) A liberdade de redistribuir cópias do programa de modo que você possa ajudar ao seu próximo; (4) A liberdade de modificar (aperfeiçoar) o programa e distribuir estas modificações, de modo que toda a comunidade se beneficie

[3] GPL é uma sigla para "General Public Licence", a licença criada por Richard Stallman, o precursor do Movimento Software Livre e fundador da FSF ("Free Software Foundation"), a principal organização referência para o movimento.

porque é mais eficiente em atrair colaboração de inúmeros desenvolvedores para o aprimoramento constante do software. A distinção entre software livre e código aberto, ainda que possa ser tratada como elemento contextual, não terá relevância central nesta pesquisa. Sempre que for me referir aos softwares livres ou de código aberto usarei o nome comumente utilizado por grande parte das comunidades de codificação social: FLOSS, que é uma junção das iniciais de "*free*", "libre" e de "*open source software*"[4]. O termo FLOSS será utilizado aqui sempre para a referência ao software livre ou ao software de código aberto ("open source").

Esta pesquisa propõe analisar as dinâmicas de autoridade e influência dentro de uma comunidade FLOSS que, contando com a participação de um movimento social, construiu um software utilizado pelo Estado Brasileiro. A participação dos movimentos sociais de economia solidária e software livre, além do uso pelo Estado Brasileiro serão elementos contextuais. O foco das análises se concentrou nas dinâmicas internas entre as organizações da comunidade, sem fazer distinção entre empresas, movimentos sociais ou o próprio Estado. A análise se dará a partir de um estudo de caso da comunidade Noosfero[5], um software criado em 2007. O Noosfero é uma plataforma web para criação de redes sociais desenvolvida em software livre e criado pela Colivre[6], uma cooperativa baiana de desenvolvimento de tecnologia. O Noosfero é um software bastante versátil que permite a criação de blogs, comunidades e empreendimentos, num contexto de plataforma de mídia social, onde as pessoas podem estabelecer relações de pertencimento (à uma comunidade ou empreendimento) e de amizade, entre perfis. O Noosfero pode ser utilizado para hospedar diversos tipos de comunidades, tanto para fins educacionais, empresariais ou exclusivamente sociais (de relacionamento ou afiliação à algum tema).

A principal motivação para o desenvolvimento do Noosfero era viabilizar uma tecnologia livre para hospedar a rede de cooperativas ligadas ao Fórum Brasileiro de Economia Solidária[7], de uma forma que as elas não tivessem que se atrelar a modelos de

---

[4] Para um explicação (em inglês) do conceito FLOSS acesse esse artigo na ngu.org https://www.gnu.org/philosophy/floss-and-foss.html

[5] Sítio oficial no Noosfero: http://noosfero.org/

[6] Sítio institucional da Colivre: http://colivre.coop.br/

[7] Segundo o Fórum Brasileiro de Economia Solidária, "A Economia Solidária pode ser definida em três dimensões: Economicamente, é um jeito de fazer a atividade econômica de produção, oferta de serviços, comercialização, finanças ou consumo baseado na democracia e na cooperação, o que chamamos de autogestão: [...] Culturalmente, é também um jeito de estar no mundo e de consumir (em casa, em eventos ou no trabalho) produtos locais, saudáveis, da Economia Solidária, que não afetem o meio-ambiente, que não tenham transgênicos e nem beneficiem grandes empresas. [...] Politicamente, é um movimento social, que luta pela

negócio baseados em softwares proprietários como era o caso das soluções que começavam a despontar na época, como o Ning[8], o Orkut ou o Facebook. Essa motivação partiu do compartilhamento de causas entre o movimento de economia solidária e o movimento software livre Brasil, que imediatamente adotou a plataforma. Para o movimento software livre Brasil, o acesso aos códigos é um princípio ético que permite aos pequenos empreendimentos se apropriarem daquele conhecimento, gerando uma atividade econômica sustentável e que não dependa de grandes empresas. Esse mesmo princípio é defendido pelo movimento de economia solidária. Dessa forma, o Noosfero nasce como um software, mas também como uma iniciativa do movimento de defesa da liberdade de acesso aos códigos e de uma economia baseada na colaboração, que contesta os modelos de negócio das plataformas de mídias sociais comerciais. Ao longo do seu desenvolvimento, o Noosfero ganhou a adesão de diversas organizações, incluindo universidades, terceiro setor e governos, permanecendo ativa até hoje.

A adoção governamental do Noosfero começa em 2011 quando a maior empresa pública de tecnologia da informação do Brasil, o SERPRO, adota o software para dar suporte ao ambiente de mídia social interno, uma espécie de intranet social com espaços de relacionamento e consultas aos funcionários da empresa. A partir dessa adoção, o SERPRO passa a oferecer soluções de tecnologia da informação baseadas em Noosfero para seus clientes de governo, iniciando o desenvolvimento de recursos específicos para consultas públicas governamentais, como a possibilidade de comentar parágrafos de um texto (Figura 1), ou estabelecer prioridades numa discussão sobre política pública (Figura 2).

mudança da sociedade, por uma forma diferente de desenvolvimento, que não seja baseado nas grandes empresas nem nos latifúndios com seus proprietários e acionistas, mas sim um desenvolvimento para as pessoas e construído pela população a partir dos valores da solidariedade, da democracia, da cooperação, da preservação ambiental e dos direitos humanos (Fonte: https://cirandas.net/fbes/o-que-e-economia-solidaria)

[8] Sítio oficial do aplicativo de mídias sociais Ning: https://www.ning.com

Figura 1: Recurso de comentário por parágrafos incorporado ao Noosfero em 2011

Fonte: Foto da tela da comunidade EGD do Participa.br (foto tirada em 2015)

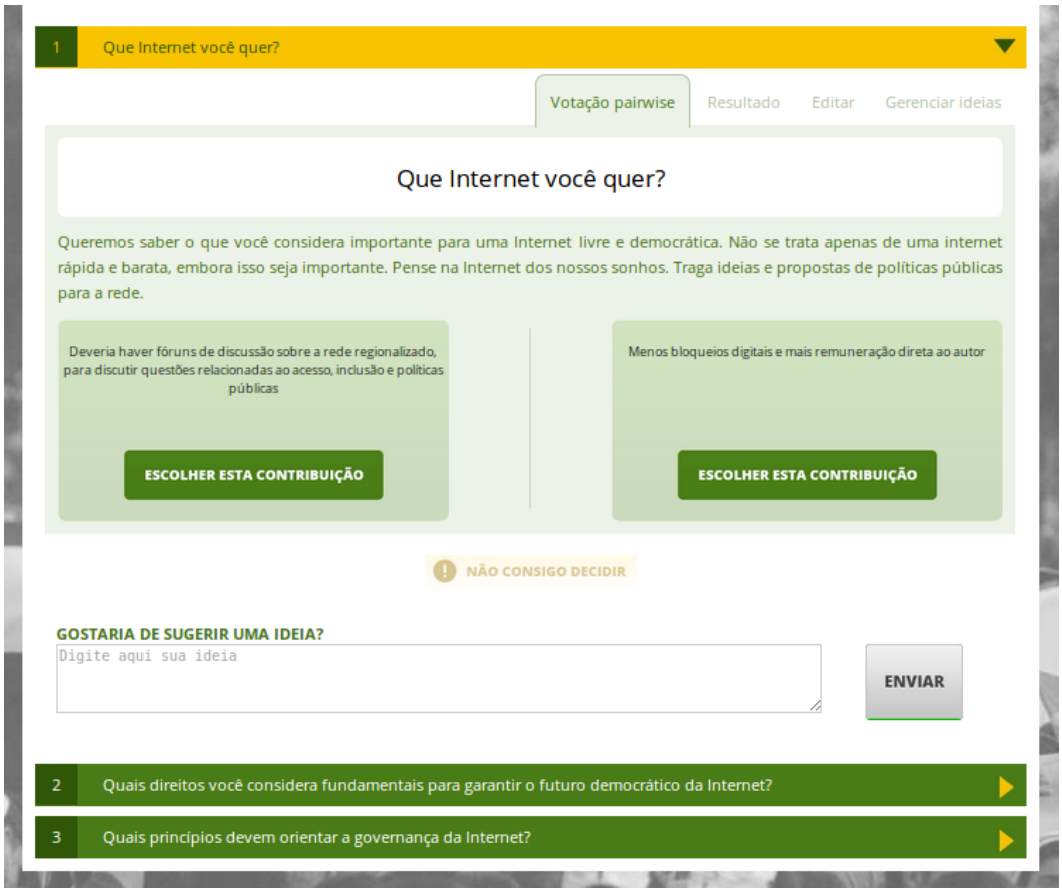

Figura 2: Recurso de priorização aos pares incorporado ao Noosfero em 2014

Fonte: Foto da tela da comunidade NetMundial do Participa.br (foto tirada em 2014)

Em 2013, interessada em utilizar uma plataforma desenvolvida em software livre com recursos de consulta pública e rede social, a Secretaria-Geral da Presidência da República adota o Noosfero como tecnologia para sua plataforma digital de participação social, o Participa.br[9]. É nesse momento que o desenvolvimento e incorporação de recursos de software voltados à participação social se intensifica e o Noosfero passa a dar suporte à política pública de participação digital do governo federal do Brasil. Além de dar suporte ao Participa.br (ativo até hoje), o Noosfero também foi utilizado em 2015 e 2016 como base da última iniciativa de participação digital do governo Dilma, o Dialoga Brasil[10].

A heterogeneidade da comunidade Noosfero, formada por diversos tipos de organizações com objetivos distintos impôs um desafio enorme à permanência da comunidade. O compartilhamento de um mesmo software por uma espécie de federação de organizações que fizeram usos tão distintos poderia muito facilmente ter levado a uma divisão

[9] A plataforma Participa.br segue em atividade no endereço http://participa.br/

[10] Até meados de 2018, a iniciativa do Dialoga Brasil seguia disponível para consulta em http://dialoga.gov.br/

da comunidade em softwares diferentes entre si e mantidos de forma independente por cada organização. Porém isso não aconteceu. A despeito disso a comunidade Noosfero se manteve unida durante toda a sua trajetória, lançando mão de diversos comportamentos e táticas para que isso fosse possível. Nessa pesquisa, estamos comparando esse feito com um ato quase heróico, uma verdadeira façanha da comunidade Noosfero. Em outras palavras, a façanha da comunidade foi essa improvável realização, onde organizações tão diferentes na sua natureza e com objetivos tão distintos entre si conseguiram permanecer unidas na mesma comunidade, compartilhando e contribuindo com o mesmo software durante seus mais de 11 anos de existência.

1.1. Justificativa da relevância do tema

O Brasil tem há alguns anos experimentado iniciativas de colaboração aberta em torno de FLOSS, na linha estratégica colocada por Sørensen & Torfing (2011), quando recomendam que "o papel dos gestores públicos não é o de produzir a inovação pública mas o de criar, institucionalizar e gerir arenas abertas e flexíveis para a interação colaborativa com outros atores relevantes ou afetados" (página 857). A primeira e mais estruturada experiência de fomento e adesão a softwares FLOSS pelo governo brasileiro se dá a partir do primeiro mandato do então Presidente Lula com um conjunto de medidas mirando "a consolidação de uma Sociedade de Conhecimento inclusiva, orientada ao desenvolvimento social, econômico, político, cultural, ambiental e tecnológico" (FREITAS & MEFFE, 2010, página 530). Parte desse esforço se inicia em 2004 com a decisão pela construção do projeto do portal do Software Público Brasileiro (SPB), que acabaria por ser lançado em 2007. A ideia central da iniciativa era ser um ambiente colaborativo que possibilitasse "o aprimoramento dos aplicativos ali disponibilizados e, conseqüentemente, a melhoria do atendimento à população" (SANTANNA, 2007, notícia). Como contam Freitas e Meffe sobre a atividade e funcionamento do Portal:

> No início do ano de 2010, o Portal do Software Público Brasileiro disponibilizava à sociedade trinta e seis soluções – ou softwares públicos. Tais soluções eram utilizadas por 66 mil usuários cadastrados à época na Rede do Portal. Em 2008 a rede crescia a uma média de 1.500 usuários por mês. Em 2009, a média já ultrapassava dois mil novos cadastros mensais. Cada software é construído, cotidianamente, por uma determinada comunidade de interesse. Cada comunidade, por sua vez, possui um

> coordenador para as contribuições feitas à produção do software e para os debates em torno de diversas questões que envolvem o artefato e a rede (como prêmios oferecidos àqueles que se destacam em sua comunidade) (FREITAS & MEFFE, 2010, página 535).

Uma pesquisa conduzida entre os anos de 2008 e 2009 buscou identificar os diferentes perfis e interesses presentes nas comunidades do portal. Segundo Freitas (2012), o portal reunia organizações com motivações sociais e políticas em torno da produção compartilhada de conhecimento junto com empresas em busca de oportunidades econômicas, ambas reunidas em torno da construção dos softwares disponibilizados ali. Além do benefício imediato da redução de custos, "observa-se também objetivos políticos claros, como o de desenvolver artefatos tecnológicos para a melhoria do atendimento à população e o de criar espaços de troca de conhecimento e tecnologia entre amplos setores da sociedade" (FREITAS, 2012, página 100). Ainda segundo a pesquisadora, atores diferentes como pessoas físicas, empresas e órgãos das diversas esferas de governo cooperam numa mesma lógica de "disponibilização de bens e serviços públicos em uma rede virtual interorganizacional de produção e difusão de conhecimento tecnológico" (Idem, página 111).

Um dos aspectos interessantes da iniciativa do portal do software público foi a sua capacidade de fortalecer a agenda de adoção, fomento e disponibilização de FLOSS pelas instituições Estatais a partir de seus resultados práticos. Para que isso fosse possível, a iniciativa acabou criando uma série de procedimentos burocráticos[11] que, mesmo tendo ajudado a fortalecer a iniciativa na disputa institucional e garantido sua duração até hoje, também a distanciou de práticas típicas de comunidades mais ativas de desenvolvimento de FLOSS na Internet. Segundo O'Maley (2013), esse diagnóstico é confirmado por um grande número de desenvolvedores brasileiros de FLOSS "que acreditam faltar à iniciativa tanto uma comunidade de programadores vibrante e inovadora quanto a oferta de software de ponta" (página 5).

O histórico do Software Público Brasileiro, para além do caráter político de defesa da causa da colaboração aberta na construção de FLOSS para melhoria das políticas e serviços públicos, revela também a necessidade de se refletir sobre as diversas maneiras que o Estado

[11] Um exemplo dos procedimentos burocráticos foi a necessidade de registrar o código fonte no Instituto Nacional de Propriedade Intelectual, além da obrigação que as comunidades utilizassem a plataforma governamental de codificação social que era tecnologicamente inferior às mantidas por empresas como Github e Gitlab, mais comumente utilizadas pelas comunidades FLOSS mundo afora

tem para aderir e impulsionar a colaboração aberta na construção de tecnologias. Ao focar essa pesquisa nas dinâmicas internas de autoridade e influência de uma comunidade FLOSS, também fazemos um convite a que os gestores públicos observem de forma mais próxima esse fenômeno e contemplem a relação do Estado com as comunidades FLOSS como parte de suas estratégias de aprimoramento da gestão das políticas públicas.

Isso se torna especialmente relevante no contexto de que a digitalização dos serviços públicos é um fenômeno irreversível. O Decreto 8.936, de 19 de dezembro de 2016,[12] instituiu a Plataforma de Cidadania Digital, onde serão disponibilizados os serviços digitais do governo federal aos cidadãos Brasileiros. Ainda que essa tendência já tenha começado há alguns anos através das iniciativas de padronização da carta de serviços e o catálogo de serviços nos governos Lula e Dilma, esse decreto se diferencia ao estabelecer prazos para lançamento da plataforma e para o cadastramento de serviços pelos órgãos, bem como para o lançamento de ferramentas de avaliação e monitoramento. Dessa maneira, é possível constatar, tanto nacional como internacionalmente, um movimento em direção à digitalização dos serviços, mesmo que os Governos demonstrem alguma dificuldade em acompanhar o ritmo do mercado (GERMANI, 2016, página 1). Entrevistado pelo portal Convergência Digital em outubro de 2017, o então diretor do Departamento de Governo Digital do Ministério do Planejamento, Wagner Araújo, afirmou que já foram catalogados 600 serviços públicos - só do governo federal - mas que a estimativa é que existam aproximadamente 2 mil serviços[13]. Diante do fato de que esses serviços estão em processo de digitalização, diversos tipos de softwares estão sendo adotados para viabilizar a oferta dos serviços e, consequentemente, dar suporte às políticas públicas. Segundo Germani (2016),

> serviços digitais se concretizam na forma de software que, acessíveis a partir de dispositivos conectados à rede, podem atender milhares – ou milhões – de pessoas simultaneamente. Softwares, por sua vez, são conjuntos de instruções, escritos em forma de texto, que indicam para o computador o que ele deve fazer. Logo, serviços digitais são, também, produtos de produção intelectual – e, portanto, conhecimento - assim como textos, leis, partituras, etc. Este entendimento é importante pois, ao analisar as novas práticas e metodologias de desenvolvimento de serviços digitais, analisaremos, sobretudo, sob a perspectiva das mudanças estruturais trazidas pelos meios digitais de comunicação, em especial a Internet, sobre a forma como a humanidade produz conhecimento (página 5).

[12] Link para o decreto: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/decreto/D8936.htm
[13] Link para a notícia citada: http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infoid=46522&sid=154

Dessa forma, muitas das decisões ligadas ao desenvolvimento dos softwares que viabilizam a oferta de um serviço público aos cidadãos passam por decisões políticas sobre a melhor maneira de apresentar as informações, quantos passos serão necessários para a finalização do serviço, qual a demanda de recursos cognitivos para compreender as informações ali colocadas e finalmente como coletar e armazenar as informações pessoais dos cidadãos. Decisões sobre essas tecnologias acabam refletindo ou condicionando as decisões sobre as políticas públicas que esses serviços suportam. Essas duas camadas estão intrinsecamente ligadas.

Ainda no que se refere aos impactos dos processos de colaboração aberta ligados ao FLOSS e suas comunidades nas Instituições Estatais, algumas pesquisas têm se debruçado em explicar o papel das TICs no aumento da transparência do Estado e abertura para participação, constituindo uma espécie de transição do modo de governo eletrônico para o da governança digital. Vaz (2017) alerta que existe um processo global de superação do modo "broadcasting" da governança eletrônica, que adquire um caráter cada vez mais relacional. Segundo o autor,

> O quadro de transformações tecnológicas e a sua interação com os processos sociais permitem que se considere a emergência de uma segunda geração da governança eletrônica. Supera-se o modo broadcasting de governança eletrônica: quebra-se o monopólio do Estado sobre as decisões e iniciativas de transparência e participação nas políticas públicas. Surgem outras formas de promover a participação, a transparência e o controle social das políticas públicas. As práticas de desenvolvimento compartilhado e os dados governamentais abertos permitem a coprodução e a produção descentralizadas de aplicações e serviços de base tecnológica. Nesse contexto, a sociedade civil e o setor privado podem passar a apropriar-se de dados públicos para produzir novas informações e serviços. Isso significa que podem emergir, desvinculadamente dos governos, formas de participação e intervenção nas decisões das políticas públicas baseadas na tecnologia (VAZ, 2017, página 91).

O ponto central desse argumento é o que traz Colombo (2006), afirmando que as TICs

> facilitariam transição de uma forma de governo tradicional a uma forma de governo relacional denominada governance, incorporando-se a um sistema complexo que introduz a participação de vários atores no governo e envolve mudanças nos papéis do poder público, e a posição que adota nos processos de governo e na utilização de novos instrumentos de governo. Neste marco de governança eletrônica se potencializam formas de participação eletrônica diversas (página 6).

Em outras palavras, o que esses autores estão dizendo é que as TICs estão transformando o contexto da participação nas políticas públicas, através de plataformas de ativismo, consultas ou outros espaços digitais onde esse processo passa a acontecer cada vez mais na velocidade da Internet. As plataformas de codificação social certamente estão inscritas nessa transformação.

Partindo da premissa que decisões sobre tecnologia afetam as políticas públicas e que existe um movimento de adoção de softwares FLOSS por governos no mundo todo, surge a necessidade de entender melhor como funcionam as dinâmicas internas dessas comunidades, que tendem a ganhar cada vez mais centralidade política. Estudar o caso concreto de uma comunidade FLOSS Brasileira, analisando os comportamentos dos agentes e as dinâmicas de influência e autoridade pode ser relevante para entender mais detalhadamente como operam esses processos de colaboração aberta, especificamente de codificação social.

2. Discussão da literatura

Alguns autores, como Castells (2008) e Benkler (2006), defendem a ideia de que estamos vivendo a emergência de uma sociedade em rede que modifica as possibilidades da associação e colaboração entre sujeitos políticos. Um caso bastante citado é o da incorporação dos princípios da criação da internet, fortemente ligados à cultura hacker, no desenvolvimento de novas formas organizacionais motivadas pela adoção da internet e da web pelas sociedades humanas (CASTELLS 2003, SHIRKY 2012, WEBER 2004). Segundo Castells (2003), é próprio da cultura hacker um processo de interação liberta, imerso numa "cultura de criatividade intelectual fundada na liberdade, na cooperação, na reciprocidade e na informalidade" (página 45). Dessa forma, os hackers "não dependem de instituições para sua existência intelectual, mas dependem, efetivamente, de sua comunidade autodefinida, construída em torno de redes de computadores" (Idem, página 43). É próprio da cultura hacker a "premência de reinventar maneiras de se comunicar com computadores e por meio deles, construindo um sistema simbiótico de pessoas e computadores em interação na Internet" (Idem, página 45).

No processo de ocupação da internet pelas comunidades virtuais, Castells indica que os valores compartilhados por essas culturas trazem muita relação com os princípios da internet. O pesquisador diz que as comunidades virtuais,

> trabalham com base em duas características fundamentais comuns. A primeira é o valor da comunicação livre, horizontal. A prática das comunidades virtuais sintetiza a prática da livre expressão global, numa era dominada por conglomerados de mídia e burocracias governamentais censoras. Essa liberdade de expressão de muitos para muitos foi compartilhada por usuários da Net desde os primeiros estágios da comunicação on-line, e tornou-se um dos valores que se estendem por toda a Internet. O segundo valor compartilhado que surge das comunidades virtuais é o que eu chamaria de formação autônoma de redes. Isto é, a possibilidade dada a qualquer pessoa de encontrar sua própria destinação na Net e, não a encontrando, de criar e divulgar sua própria informação, induzindo assim a formação de uma rede (CASTELLS, 2003, página 48).

Por outro lado, os princípios da internet e a cultura hacker não são suficientes para manter uma cultura de compartilhamento e livre acesso nessas comunidades. Segundo Kranich (2016), o problema fundamental do conhecimento na rede se reduz ao problema da captura digital, tanto por Estados autoritários quanto pelas grandes corporações. A autora define essa captura como uma espécie de novo cercamento que está colocando em risco o acesso da humanidade ao conhecimento. Segundo Laval & Dardot (2014) temos que considerar que a informação disponível na rede - definida como “Comum de Conhecimento” pelos autores que estudam o Comum (Hess & Ostrom, 2016 / Laval & Dardot, 2014) - são mais vulneráveis do que se possa acreditar.

> A centralização em sítios privados e públicos aumenta o temor da desaparição dos recursos de informação. Mas, sobretudo, a criação de cercamentos pelo mercado e pelo governo, como Boyle ou Bolier tem evidenciado, pode levar a um fechamento do acesso ao conhecimento e um esgotamento dos fluxos de informação (Laval & Dardot, 2014, p. 187).

Dessa forma, ainda que a auto organização de comunidades que aproveitaram as possibilidades de trabalho comum oferecidas pelas novas tecnologias da internet tenham sido crucial para o surgimento dos comuns de conhecimento (com o FLOSS incluído aí), elas se transformaram em instituições que atuam sob um conjunto de regras que permitem o funcionamento dessas comunidades para que o comum que elas mantém não se esgote (Laval & Dardot, 2014, p. 188). Se por um lado a popularização dos computadores e a conectividade da internet acabaram removendo barreiras para a

> desconcentração dos meios de comunicação e produção da informação, nada garante que essa condição técnica essencial para a construção de um novo espaço público seja suficiente para garantir o surgimento de uma informação democrática e de uma cultura surgida da produção comum (Laval & Dardot, 2015, p. 211).

Diante disso é fundamental recusar os profetismos em torno da tecnologia digital como um vetor inexorável para uma sociedade livre. Segundo os autores, "outros esquemas de utilização das tecnologias podem servir para estratégias muito diferentes e induzir relações sociais muito distintas" (Laval & Dardot, 2014, p. 211). Essa é a batalha que está colocada no campo das novas tecnologias.

Seguindo na demonstração desse contraponto à uma visão romantizada das comunidades virtuais, os trabalhos de von Bülow (2018) e Gerbaudo (2016) oferecem um diagnóstico sobre as dinâmicas internas dos movimentos sociais quando se apropriam de plataformas digitais na internet. O trabalho de von Bülow (2018), que analisou o impacto nas relações de poder internas aos movimentos estudantis chilenos no processo de apropriação das mídias sociais, nos permite, a partir de seus achados, olhar de perto como funcionam as dinâmicas organizacionais dos movimentos e sua relação com o uso do twitter. Esse trabalho nos ajuda a resgatar alertas clássicos sobre os perigos inerentes ao que Philips chamou de "paradoxos da participação", quando a baixa formalização de lideranças leva a pouca responsabilização das pessoas que estão atuando como líderes e elites de fato (PHILIPS, 1991). A autora também menciona argumentos similares sobre os perigos da invisibilidade das lideranças, tanto em Jo Freeman (1970) quanto em Melucci (1996).

Complementarmente, Gerbaudo (2016) faz coro a esse diagnóstico ao dizer que a narrativa de horizontalidade e ausência de liderança dos movimentos de protestos baseados em mídias sociais é politicamente perigosa, pois cria uma cortina de fumaça na frente do que realmente está acontecendo no interior dos grupos de vanguarda do movimento, fazendo com que as reais lideranças permaneçam sem nenhum tipo de controle e prolifere a briga de facções (GERBAUDO, 2016, páginas 10-11). A partir de seus achados de pesquisa, Gerbaudo (2016) demonstra que a atuação dos grupos responsáveis por gerir os perfis de mídias sociais dos novos movimentos de protestos se assemelha com a função das vanguardas políticas que, gerindo os poderosos canais de comunicação do movimento, operam o direcionamento político e estratégico da ação coletiva. Sobre a questão da hierarquia e do poder, Gerbaudo (2016) é convincente em mostrar como os valores da cultura hacker (que ele chama de

tecno-libertários) encontram pouca aderência nas práticas organizacionais dos movimentos de protesto baseados em mídias sociais quando observadas mais de perto e analiticamente (página 8). O ideal de abertura se contrapõe com a necessidade de estabelecer um direcionamento estratégico e um nível mínimo de proteção na definição das mensagens que serão publicadas pelo grupo. O ideal de horizontalidade contrasta com as diversas camadas de poder nas relações entre os membros do grupo e a própria arquitetura tecnológica das ferramentas com uso de senhas e diferentes papéis para os perfis. E por fim, o ideal de não-liderança contrasta com o papel exercido pelas vanguardas no direcionamento da ação coletiva do movimento (Idem, página 15).

Essas pesquisas apresentadas tiveram como objeto as plataformas de mídias sociais utilizadas para a mobilização política, com foco na sua apropriação por organizações e indivíduos da sociedade civil. O uso de mídias sociais aplicadas ao trabalho colaborativo para a construção de software de código aberto também tem sido objeto de atenção pela literatura. Os ambientes de desenvolvimento colaborativo se diferem das mídias sociais pela mudança de seu eixo central de organização. Enquanto estas são organizadas em torno dos laços de amizade e interesse entre usuários, aqueles são fundamentalmente organizados em torno do código mantido pelas comunidades. Badashian, Esteki e Gholipour (2014) explicam que, “nas redes de desenvolvimento de software, os indivíduos são conectados principalmente através e ao redor do código (ou seja, projetos, perguntas, etc.), enquanto em redes sociais típicas, as conexões são criadas diretamente entre usuários (ou seja, "amizade", "seguir" etc.)” (página 2, tradução nossa). Por outro lado, ainda que o foco principal do trabalho nessas plataformas seja a colaboração em código, há diversos aspectos sociais que orientam o processo decisório nas comunidades ali presentes. Contrariando o senso comum entre os desenvolvedores de que toda decisão é técnica e baseada exclusivamente na qualidade do código, Tsay, Dabbish e Herbsleb (2014) citam alguns trabalhos sobre FLOSS com sugestões de que a decisão sobre as contribuições é carregada de questões sociais e baseada “num processo mais nuançado que envolve outros fatores além do mérito técnico” (página 1, tradução nossa). Segundo as autoras, as dinâmicas sociais vão muito além do código, sendo possível considerar um projeto FLOSS como uma comunidade virtual (online).

> As comunidades on-line estão centradas em contribuições de uma grande variedade de usuários. As comunidades on-line bem-sucedidas contam com membros que contribuem com seus recursos exclusivos para a comunidade, como usuários que carregam vídeos no YouTube ou postam fotos ou comentários no reddit. Kraut e Resnick analisam os desafios que as

> comunidades on-line enfrentam ao tentar incentivar a contribuição: combinar os usuários com as contribuições necessárias, fazer pedidos aos membros, usar motivadores intrínsecos e extrínsecos e agrupar usuários. Eles analisam as evidências que demonstram que o feedback constante aos membros, sejam os níveis de personagem no World of Warcraft ou comentários da comunidade no YouTube, motiva os membros a criar mais contribuições. Da mesma forma, combinar contribuições com contato social também incentiva novas contribuições (TSAY, DABBISH & HERBSLEB, 2014, página 2, tradução nossa).

O principal achado dos autores consistiu em perceber que, para ter suas contribuições aceitas, os desenvolvedores tinham que passar por um processo de socialização, além de usar diferentes métodos para influenciar os resultados. Foram identificados três métodos utilizados pelos membros. O primeiro foi o da "pressão pela audiência", quando o responsável pela contribuição mobiliza outros atores para convencer os desenvolvedores responsáveis pela base de código da comunidade (chamados de desenvolvedores "core" ou "commiters") a aceitarem suas contribuições. Essa pressão é feita tanto diretamente aos membros do grupo que toma as decisões quanto via manifestações públicas de apoio nas ferramentas da plataforma colaborativa de desenvolvimento (Idem, página 7-8). Essas manifestações públicas são apontadas pelas autoras como um tipo específico de pressão, que é definido como "suporte da comunidade". Por fim, um terceiro método seria acionar diretamente algum membro do grupo central da comunidade. Na pesquisa isso era feito diretamente pela plataforma de codificação social, utilizando o recurso da "@" para mencionar de forma específica esse usuário. Por fim, as autoras concluem que esses métodos são viáveis pela razão de que nesses ambientes abertos de desenvolvimento colaborativo de FLOSS, "uma maior variedade de partes interessadas pode influenciar os requisitos do projeto de software através dos espaços de discussão disponíveis" (Idem, página 10).

A organização social responsável pela construção e gestão de um software FLOSS é a comunidade, cuja conformação é fortemente influenciada pelo tipo de software que desenvolve. NAKAKOJI, YAMAMOTO & NISHINAKA, (2002) oferecem uma tipologia de projetos com base nas características inerentes ao software, classificando-os em três tipos. O primeiro é o FLOSS orientado a exploração. Nesse tipo de projeto, a cultura é similar a da pesquisa científica, onde o objetivo central é compartilhar resultados científicos. A expectativa de qualidade é alta e o principal objetivo é transmitir conhecimento e permitir o reuso do código, "permitindo a outros avançar subindo nos ombros do desenvolvedor anterior

pelo reuso do código aberto" (NAKAKOJI, YAMAMOTO & NISHINAKA, 2002, p. 82). Devido a sua característica autoral, o controle exercido pelas lideranças tende a ser forte e só são aceitas contribuições que coadunam com a visão da liderança. São exemplos desse tipo de software as ferramentas usadas pelos desenvolvedores para construir outros softwares, como os compiladores e as linguagens de programação.

No segundo tipo, o FLOSS orientado à utilidade, os desenvolvedores normalmente modificam softwares existentes para resolver uma necessidade específica. Normalmente são projetos mais preocupados com aspectos operacionais (em outras palavras, feitos simplesmente para funcionar) do que com a qualidade do código fonte. São exemplos desse tipo de softwares, os "drivers de hardware", que tem o objetivo de fazer os computadores funcionarem para os usuários, incluindo coisas como redes wifi e as impressoras. Geralmente são softwares sem muita contribuição ou liderança pois normalmente fazem parte de comunidades maiores.

E finalmente o terceiro tipo de projeto, que também é o que tende a ser mais conhecidos por usuários não técnicos é o FLOSS orientado a serviço. Esses softwares oferecem serviços completos, ou seja, softwares que são utilizados por organizações não técnicas e usuários finais. Softwares para redes sociais e blogs são exemplos desse tipo de projeto. As comunidades tendem a ser grandes e com diversos tipos de organização. São nesses projetos que as formas de governança são mais desenvolvidas. Todos os três tipos de projeto FLOSS fazem uso dos ambientes de codificação social, porém com requisitos distintos. No caso do Noosfero, um projeto FLOSS orientado a serviço, veremos mais adiante como essa característica e outras específicas do próprio Noosfero foram fundamentais para o tipo de organização federada que se formou em torno dele.

Dentro do tema dos processos de socialização que influenciam nos resultados de incorporação de código pela comunidade, NAKAKOJI, YAMAMOTO & NISHINAKA, (2002) oferecem referências importantes que vamos utilizar para criar uma tipologia de duas táticas utilizadas pelos líderes de comunidade FLOSS. Essas táticas não tem objetivo explícito de influenciar no processo decisório da comunidade mas sim contribuir para a longevidade da mesma. Porém seu efeito acaba sendo o de aumentar o nível de porosidade da comunidade, contribuindo para que a atuação dos desenvolvedores periféricos influencie na decisão de incorporação de código.

A primeira dessas táticas é a de encorajar os membros a progredir em direção ao centro da comunidade a partir de suas contribuições. Segundo os autores,

> Os líderes de projeto e membros centrais de uma comunidade FLOSS não devem focar apenas na evolução do software propriamente dito, mas também se esforçar para criar um ambiente e cultura que estimule o senso de pertencimento à comunidade e mecanismos que encorajem novos membros a se mover em direção ao centro da comunidade FLOSS através de suas contribuições contínuas (NAKAKOJI, YAMAMOTO & NISHINAKA, 2002, p. 82).

Essa tática portanto, ajuda a definir toda ação relacionada à manutenção de uma cultura e ambiente propício a contribuição, principalmente dos membros periféricos. Essa necessidade, que está muito mais ligada a garantir a sobrevivência da comunidade no longo prazo, acaba tendo o efeito de suavizar a autoridade dos desenvolvedores centrais que buscam olhar de forma apreciativa para toda contribuição dos outros membros, de modo a não desencorajar novas contribuições.

A segunda tática diz respeito a se adequar às necessidades dos membros para evitar a fragmentação da comunidade. O livre acesso e uso do código fonte por qualquer dos membros faz com que o recurso do "fork" esteja sempre disponível para todos e seja utilizado caso um grupo de membros deixe de se sentir contemplado pelas decisões tomadas no grupo central de desenvolvedores. Dessa forma, as lideranças de uma comunidade FLOSS estão sempre convivendo com o fantasma da fragmentação da comunidade, e a principal maneira de afastá-lo é se adequando às necessidades dos diversos membros e grupos que fazem parte da comunidade, num trabalho de concertação entre vários interesses e comunicação clara das justificativas de cada decisão.

Dentro dessa mesma temática, não é de hoje que os estudos da Computação e Engenharia de Software sobre desenvolvimento colaborativo analisam o fenômeno do trabalho nos ambientes digitais para o desenvolvimento de FLOSS. Reconhecendo as profundas transformações nas fronteiras do trabalho de desenvolvimento proporcionadas pelos ambientes de codificação social, Dabbish & Stuart (2012) demonstram que os mecanismos de transparência presentes na plataforma do GitHub viabilizam o trabalho cooperativo, diminuindo a necessidade de encontros sincronizados. Segundo as autoras, seus achados trazem informações relevantes sobre o "desenho desse tipo de mídia social para colaboração em larga escala e implicam uma variedade de maneiras pelas quais a

transparência pode apoiar inovação, compartilhamento de conhecimento e construção de comunidade" (DABBISH & STUART, 2012, página 1286). McDonald & Goggins (2013) também perceberam o valor da transparência e das ferramentas sociais de colaboração no dia a dia dos desenvolvedores que operam nesses ambientes. Segundo alguns resultados de suas pesquisas, também com base no GitHub, os desenvolvedores associam a experiência de uso da plataforma com o desenvolvimento de práticas mais democráticas e transparentes de trabalho. O ambiente de discussão em torno das contribuições permite que haja um diálogo público atrelado ao processo decisório sobre novas incorporações de código, permitindo que quem proponha a alteração tenha acesso e exponha todos os argumentos e todos os interessados naquela mudança possam se manifestar (MCDONALD & GOGGINS, 2013, página 142). O uso do mecanismo das menções (@usuário), típico dos ambientes de mídias sociais, também é utilizado como instrumento de trabalho para aumentar a agilidade na resolução dos problemas e melhorar a comunicação entre os desenvolvedores e usuários (ZHANG, WANG & YIN, 2015, página 4). O histórico público desse processo (transparência) também permite que ele seja posteriormente conhecido por qualquer desenvolvedor ou usuário da comunidade. A coordenação do trabalho requer pouca comunicação síncrona (reuniões) já que é baseada na transparência e na responsabilidade de cada membro em manter atualizado o status do seu trabalho (KALLIAMVAKOU, DAMIAN & SINGER, 2014, página 7). Além disso,

> a visão de colaboração centrada no código vai em uma direção diferente da visão mais tradicional da colaboração como trabalho conjunto, mas que, no entanto, mostra sinais claros de benefícios. A inspiração para esses princípios de trabalho independente e auto-organizado está fortemente relacionada à mentalidade do ambiente de desenvolvimento FLOSS e ao uso de ambientes de codificação social para gerenciar uma base de código colaborativa (Idem, página 10).

As ferramentas disponibilizadas para as comunidades FLOSS nesses ambientes de codificação social fomentam uma cultura de colaboração atrelada a elas. Em um trabalho que pesquisou as percepções dos membros dessas comunidades, Vasilescu, Filkov e Serebrenik (2015) descobriram que os desenvolvedores trabalham com uma visão ampliada sobre quem são seus colegas de trabalho, considerando como parte da equipe qualquer pessoa que faz alguma contribuição no repositório da comunidade (página 5). Além disso, encontraram um resultado bastante positivo em relação à percepção sobre diversidade: para a maioria dos

entrevistados, a diversidade de membros nas comunidades contribui para “fornecer novas ideias, perspectivas, habilidades e abordagens para resolver problemas” (VASILESCU, FILKOV & SEREBRENIK, 2015, página 7).

Estudar as comunidades de construção colaborativa de FLOSS também foi tarefa de diversas pesquisas no campo da Educação e Teoria das Organizações. Segundo os resultados dessas pesquisas, as comunidades FLOSS possuem alguns papéis que costumam se repetir. Para Martinez-Torres & C. Diaz-Fernandez (2014),

> embora uma estrutura hierárquica rigorosa não exista nas comunidades FLOSS, a estrutura das comunidades não é completamente plana. Cada comunidade possui uma estrutura única dependendo da natureza do software e da sua população membro. A estrutura de uma comunidade FLOSS difere na porcentagem de cada papel em toda a comunidade. Nesse sentido, os membros de uma comunidade de FLOSS assumem determinados papéis por eles próprios de acordo com seu interesse pessoal no projeto, em vez de serem designados por outra pessoa. As classificações anteriores distinguem as seguintes oito funções: Líder de projeto; Membro principal, Desenvolvedor ativo, Desenvolvedor periférico, Bug Fixer, Bug Reporter, leitor e usuário passivo. As influências que os membros têm no sistema e na comunidade são diferentes, dependendo dos papéis que desempenham (página 58).

Segundo Martínez-Torres (2012) a estrutura de uma comunidade FLOSS só pode ser identificada pela atividade dos seus membros,

> sendo os nós da rede os membros da comunidade, enquanto os arcos representam o fluxo de interações entre usuários. [...] Esta variedade de perfis de usuários leva a uma estrutura núcleo / periferia, onde o núcleo do grupo de desenvolvedores está localizado no centro da comunidade e o resto deles longe do centro, dependendo de suas contribuições (página 1).

Uma das características mais distintivas do FLOSS é o fato de que a sua produção conta com a participação de voluntários do mundo todo, executando papéis de acordo com a sua vontade e disponibilidade (Idem). Nesse sentido, não é uma hierarquia pré-definida que define o que esses voluntários vão fazer. Ainda que haja um grupo de desenvolvedores restrito com acesso à incorporação de atualizações no código, não são eles que definem como os outros desenvolvedores vão trabalhar. Ainda segundo a autora, é devido a essa fluidez, que a estrutura das comunidades de FLOSS só podem ser estudadas a partir da atividade dos seus membros e não a partir de uma estrutura pré-definida.

A partir da constatação dessa diferença de atuação entre os membros de uma comunidade FLOSS - o que Kuk e Martínez-Torres vão chamar de “desigualdade de

participação" (KUK, 2006; MARTÍNEZ-TORRES, 2012) - os diversos papéis podem ser classificados em três grupos: desenvolvedores centrais, desenvolvedores ativos e desenvolvedores periféricos. Conforme explica Martínez-Torres,

> O membros centrais são responsáveis por guiar e coordenar o desenvolvimento de um projeto FLOSS. Eles usualmente permanecem envolvidos com o projeto durante um longo período de tempo tendo feito contribuições significativas ao desenvolvimento e evolução do sistema. Moderadores e líderes estão incluídos nesse grupo. Os desenvolvedores ativos [outro grupo] já são aqueles membros da comunidade que fazem contribuições regulares ao projeto. Finalmente, os desenvolvedores periféricos contribuem ocasionalmente com novas funções para o software existente. Essa contribuição é irregular, e o período de envolvimento é curto e esporádico. Os "free riders" (pessoas que estão apenas buscando respostas sem fazer nenhuma contribuição) também estão incluídas nesse grupo (MARTÍNEZ-TORRES, M. R, 2012, página 1).

Essas pesquisas têm demonstrado que as comunidades FLOSS normalmente possuem um grupo de desenvolvedores principais ("core group" ou "commiters") que possuem maior influência e são responsáveis por um papel de intermediação com o resto dos membros. A forma de participação nas comunidades se dá da periferia para o centro. Explica Martínez-Torres (2012) que quem não faz parte da comunidade "só pode adquirir autoridade dentro do projeto através do procedimento legítimo de participação periférica" (Idem, página 9). Apesar dessa desigualdade na participação, os rumos da comunidade e do software não são definidos a priori por essas lideranças, mas vão se delineando a partir do processo colaborativo entre usuários e desenvolvedores (NAKAKOJI, YAMAMOTO & NISHINAKA, 2002, página 76). Estudos que buscaram entender as comunidades FLOSS para além do software, identificaram a importância da participação periférica.

> Por um lado, eles [os participantes periféricos] representam os futuros contribuidores centrais, pois aumentarão seus conhecimentos através da interação com os contribuidores regulares. Por outro lado, representam uma medida referente ao interesse dos usuários no projeto de software mantido pela comunidade. Se um projeto FLOSS não é capaz de atrair pessoas para os fóruns de discussão, isso significa que o software tem um interesse limitado entre usuários potenciais e, finalmente, o projeto será abandonado (TORAL, MARTÍNEZ-TORRES & BARRERO, 2009, página 11).

Uma explicação para a dinâmica centro/periferia das comunidades FLOSS pode ser encontrada na teorização que DAHLANDER & O'MAHONY (2011) apresentam sobre o conceito de autoridade lateral. Os autores relacionam esse tipo de autoridade aos arranjos onde a progressão do indivíduo ocorre no contexto de organizações planas:

> A noção de progresso sem hierarquia pode parecer paradoxal: Como podem os indivíduos avançar em organizações que são relativamente planas? Nós desmembramos esse paradoxo mostrando que a progressão sem hierarquia pode ser obtida quando indivíduos "progridem em direção ao centro" (tocando mais áreas do projeto) em oposição a "subir na carreira" (gerindo mais indivíduos) (DAHLANDER & O'MAHONY, 2011, p. 961, tradução nossa).

Com isso, os autores operacionalizam o conceito de autoridade lateral para estudos empíricos da seguinte forma: "nós definimos ele [o conceito de autoridade lateral] como autoridade sobre o trabalho coletivo que não inclui autoridade vertical sobre indivíduos" (Idem, p. 962, tradução nossa). Em outras palavras, quando a autoridade lateral dos indivíduos aumenta, isso implica que eles ganham responsabilidade e poder de decisão (ou direito de decidir) sobre uma maior parcela do trabalho sem que isso implique passar a supervisionar o trabalho de ninguém (Idem, p. 962).

Com base na literatura que estuda comunidades oriundas do contexto corporativo, os autores identificaram comportamentos que contribuíram para essa progressão em direção ao centro. Segundo eles, esses comportamentos são relevantes para estudos sobre comunidades voltadas à coordenação de trabalho intelectual ("knowledge work"): "O grau em que os indivíduos engajarem em (1) contribuições técnicas, (2) comunicação técnica e (3) trabalho de coordenação" (Idem, p. 963) pode prever o quanto de autoridade lateral eles vão obter ao longo do desenvolvimento do trabalho coletivo.

O primeiro comportamento, as contribuições técnicas, dizem respeito a realizar tarefas ligadas ao principal produto do trabalho realizado pelo grupo. No caso de uma comunidade de software, essa tarefa consiste em escrever código. Essa atividade tem enorme importância nesse tipo de comunidade, pois sua sobrevivência depende da existência de pessoas dispostas a realizar o trabalho. Então é quase óbvio que quem se empenha na realização deste trabalho irá aumentar sua reputação. Além disso, um indivíduo que realiza muitas tarefas começa a ser visto como alguém que conhece mais o trabalho coletivo do que outros, aumentando sua legitimidade para tomar decisões sobre ele.

O segundo comportamento, a comunicação técnica, diz respeito ao segundo componente do processo de aprendizado que é o compartilhamento do conhecimento através da conversação. Por mais que as comunidades de conhecimento, especialmente as comunidades FLOSS, garantam acesso praticamente irrestrito de todos ao trabalho coletivo,

muitas vezes é nas explicações que o aprendizado é potencializado. Como essas conversações são públicas, aqueles que engajam mais em responder e explicar tendem a ganhar mais respeito dos pares e consequentemente progredir em relação à autoridade lateral.

Por fim, o terceiro dos comportamentos, o trabalho de coordenação, diz respeito a gerir o trabalho coletivo para que ele atinja um objetivo comum à comunidade. No contexto de comunidades espalhadas geograficamente, o trabalho de coordenação é ainda mais desafiador. As plataformas de codificação social oferecem diversos recursos de coordenação, mas o resultado do trabalho coletivo ficaria estagnado sem esse tipo de trabalho. É uma oportunidade para que certas lideranças possam brilhar. Quando o trabalho de coordenação leva todos a um resultado satisfatório, o respeito pelos pares aumenta muito. Por isso faz sentido que esse comportamento seja responsável pela progressão ao centro da comunidade, ou seja, pelo aumento da autoridade lateral do indivíduo.

Ainda que essa forma de colaboração aberta via codificação social não tenha sido suficientemente estudada pela Ciência Política, foi o politólogo Steven Weber (2004) o pioneiro em explorar a conexão da cultura hacker com as práticas organizacionais desenvolvidas pelas comunidades de desenvolvimento colaborativo de software de código aberto. Segundo Weber (2004), o processo de desenvolvimento do FLOSS é um sistema de governança baseado no direito de distribuir[14] (WEBER, 2004, página 228). O pesquisador também afirma que o processo do código aberto constrói novas possibilidades no sentido de que há uma organização política operando a gestão de conflitos, poder, interesses, regras, normas comportamentais, procedimentos de tomada de decisão e mecanismos de sanção. Mas essas comunidades (FLOSS) não são uma organização política que parece familiar com a lógica da economia política da era industrial (Idem, página 3). Não é um processo "caótico e desregrado no qual todos têm igual poder e influência" (Idem, página 3), nem uma comunidade idílica de amigos com a mesma opinião onde reina o consenso e os acordos fáceis. De fato, o conflito é usual nessas comunidades (FLOSS); é endêmico e inerente ao processo do open source (Idem, página 3). Ainda segundo Weber, o processo do "open source" demonstra a "viabilidade de um sistema de inovação massivamente distribuído que

[14] O movimento de software livre, através da "Free Software Foundation" advoga por licenças que garantem as 4 liberdades básicas, que são (1) A liberdade de executar o programa, para qualquer propósito; (2) A liberdade de estudar o programa, e adaptá-lo para as suas necessidades; (3) A liberdade de redistribuir cópias do programa de modo que você possa ajudar ao seu próximo; (4) A liberdade de modificar(aperfeiçoar) o programa e distribuir estas modificações, de modo que toda a comunidade se beneficie.

estende as fronteiras das noções convencionais sobre os limites da divisão de trabalho" (Idem, página 14). Essas características ficam claras quando diz que as comunidades de código aberto "descrevem uma estrutura nascente de cultura e comunidade", incluindo critérios "para entrar (e sair), papéis de liderança, relações de poder, questões distributivistas, formas de educação e socialização" (Idem, página 15). O trabalho de Weber nos provoca a pensar o fenômeno do desenvolvimento colaborativo de FLOSS como algo não restrito a essas comunidades mas como uma inovação social com impactos maiores em outras áreas da sociedade.

Complementarmente, é na teoria sobre Autoridade Prática da Ciência Política (ABERS & KECK, 2013) que vamos buscar o vocabulário teórico para a tarefa de sistematizar a agência dos indivíduos quando buscam influenciar as regras e as decisões da comunidade ("institucional change"), conceitualizando melhor o que entendemos por autoridade. Vamos usar o conceito de autoridade prática na análise da agência individual dos atores. Falar em comportamentos de autoridade lateral e táticas de influência, é falar também no papel criativo dos agentes de criar e utilizar recursos para obtenção desse objetivo. E esse papel criativo é direcionado à questionar ou reforçar as regras vigentes na comunidade, dentro do que as autoras chamam de transformação das instituições (ABERS & KECK, 2013, p. 2). O conceito de instituições adotado por elas se aplica bem à forma que enxergamos uma comunidade FLOSS, baseada em diversas "formas comumente aceitas de fazer as coisas" (Idem, p 3, tradução nossa). As regras nas quais uma comunidade FLOSS é baseada se originam de diversas fontes construídas pela prática, como as regras criadas e mantidas pelos desenvolvedores centrais e regras inspiradas em outras comunidades FLOSS. Até mesmo as "affordances"[15] (MENDONÇA & AMARAL, 2016, p. 427) das plataformas de codificação social são fruto do consolidação das melhores práticas do desenvolvimento de software colaborativo[16]. Definir as instituições das comunidades FLOSS a partir de suas regras colocadas em prática também foi o caminho escolhido por HESS & OSTROM (2016) ao

[15] O conceito de "affordance" foi trazido por Mendonça & Amaral (2016): "Baseamo-nos, aqui, na ideia de affordance (apud Bendor, Haas Lyons e Robinson, 2012; Earl e Kimport, 2011), segundo a qual a estrutura tecnológica de um artefato induz certos comportamentos e deles participa (apud Latour, 2012).

[16] A primeira versão do GIT foi criada pelo desenvolvedor Linus Torvalds se inspirando nas melhores práticas dos sistemas existentes para versionamento de código mas principalmente inovando na capacidade de suportar contribuições distribuídas de múltiplos desenvolvedores. A motivação para sua criação foi uma necessidade do desenvolvimento do sistema operacional Linux. A criação de Torvalds foi utilizada como base para as atuais plataformas de codificação social
Referência: https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git

definir o FLOSS como um tipo de Comum de conhecimento (a obra delas tem um capítulo dedicado à discutir o arranjo das comunidades FLOSS como um arranjo de Comum). Segundo Hess & Ostrom (2016), as instituições

> são normas e regulamentos formais e informais que uma comunidade assume e aplica. O conceito de instituições que usaremos [...] não equivale automaticamente com o escrito nas disposições formais. O que um investigador deseja analisar e explicar são mais as normas estabelecidas por direitos e proibições operacionais para os indivíduos em uma situação concreta (Idem, p. 66, tradução nossa).

Assim, a ideia de instituição está no centro da definição de comum, pois ela evidencia as regras formais e informais sob as quais os atores mantém aquele recurso compartilhado. Dessa forma, ainda que essas autoras que teorizam sobre o Comum de conhecimento não trabalhem a ideia de transformação das instituições da mesma forma que Abers & Keck (2013) - já que a ideia de "crafting" de Ostrom não tem a mesma centralidade na agência -, elas concordam com a centralidade da prática na materialização das instituições. Essa convergência permite que possamos utilizar com mais segurança a teoria de autoridade prática de Abers & Keck (2013) nesse estudo sobre uma comunidades FLOSS.

Para Abers & Keck (2013), autoridade prática pode ser "entendida como um poder-em-prática gerado quando atores específicos (indivíduos ou organizações) desenvolvem capacidades e ganham reconhecimento dentro de uma área de política específica, possibilitando-os influenciar o comportamento de outros atores" (ABERS & KECK, 2013, p. 2, tradução nossa). O acúmulo de autoridade prática de um indivíduo ou organização permite que influenciem as ações de outros (Idem, p. 3). O conceito de autoridade utilizado pelas autoras quando analisam a atuação de atores fora da institucionalidade Estatal será útil para essa pesquisa. Segundo elas, o uso que fazem é consistente com a definição Weberiana de um tipo de poder que é aceito com base na legitimidade, porém nesse caso não confinado ao Estado. Da mesma forma, o poder conferido por esse tipo de autoridade é o de influenciar as ações de outros indivíduos ou organizações, porém sem utilizar os tipos de capacidades inerentemente Estatais, como o monopólio legítimo do uso da violência ou da cobrança de impostos. O tipo de capacidade utilizada para o acúmulo dessa autoridade pode ser definida como "formas de autoridade prática não exclusivas do Estado" (Idem, p. 8, tradução nossa). Na sua obra, as autoras trazem pelo menos dois exemplos de como atores não Estatais tiveram um papel central em ajudar as instituições em acumular autoridade prática, de modo que fica

evidente que esse tipo de autoridade "pode ser construída fora do Estado, sem precisar se basear nos mecanismos formais do poder Estatal" (Idem, p. 170, tradução nossa).

Além de capacidades, os atores que constroem autoridade prática também devem mobilizar o que as autoras chamam de "reconhecimento" (Idem, p. 9). O reconhecimento é necessário para que o processo de influenciar comportamentos se complete, já que os outros (que serão influenciados) precisam confirmar que a tal autoridade existe. Por isso que apenas capacidades não é suficiente para o acúmulo de autoridade prática. Mas como os atores acumulam autoridade e reconhecimento, então? Esse processo ocorre através do que as autoras chamam de "práticas de construção institucional" que podem ser resumidas em dois tipos gerais: "engajamento com outros atores e experimentação em resolução de problemas" (Idem, p. 17). O engajamento diz respeito a se conectar mais e usar essas conexões para mover ideias e recursos. A experimentação "envolve combinar e usar ideias, recursos e relacionamentos de novas maneiras" (Idem, p. 17, tradução nossa). Ambos os tipos de práticas estão interrelacionados, já que os organizadores da experimentação podem utilizar as redes para resiliência e suporte (Idem, p. 19). Em resumo, o trabalho de influenciar nas regras das instituições

> envolve dois tipos de atividades sobrepostas: engajamento e experimentação. Quando essas atividades se reforçam mutuamente, elas podem ter um efeito transformador em ideias, recursos e relacionamentos que podem potencialmente produzir novas capacidades e construir reconhecimento. Quando isso ocorre, essas organizações atingiram algum grau de autoridade prática (Idem, p. 199, tradução nossa).

Buscaremos portanto, relacionar os comportamentos e táticas para influenciar a decisão na comunidade com as práticas de construção institucional e acúmulo de autoridade prática. Nos parece que esse quadro conceitual pode trazer valiosas intuições para o contexto específico das comunidades FLOSS.

Além das práticas de construção institucional, há uma outra contribuição trazida pelas autoras que pode ser útil para essa pesquisa. Nos referimos nesse caso à ideia de "institutional entanglement" ou entrelaçamento institucional, que ocorre quando os papéis e arena de atuação das instituições não estão claros e há muito sombreamento de competências. Segundo as autoras, o caso Brasileiro da política de recursos hídricos se deve a questões históricas em que novas instituições foram surgindo sem que as velhas fossem adequadamente "aposentadas", mantendo esse entrelaçamento confuso. Além disso é comum algum nível de

entrelaçamento em se tratando de países com regime federativo, como também é o caso Brasileiro. A parte interessante dessa ideia de entrelaçamento é que ele fornece um ambiente fértil para a criatividade. Segundo as autoras,

> O entrelaçamento provavelmente provê mais oportunidades para o deslocamento espacial criativo e inovação institucional do que as configurações institucionais mais uniformes e claramente hierárquicas. O leque[17] de regras existentes, regulações, e formas de interação é bem colorido, dando margem para interpretação e bricolagem. Por outro lado, torna mais difícil transformar essas inovações em práticas duradouras, já que os recursos são dispersos e as hierarquias contestadas. [...][O esforço de transformação institucional num contexto de entrelaçamento] deixa sombras institucionais [...] que seguem existindo e mantendo algum recurso e autoridade. Esses caminhos incompletos preenchem o espaço institucional, como os muitos galhos de uma árvore, alguns dos quais nunca foram capazes gerar frutos. A imagem de galhos entrecruzados de uma árvore contrasta com as noções tradicionais de "path dependence", nas quais é presumido que os galhos nunca se encontrem depois de saírem do tronco. No sistema entrelaçado que estudamos, há muitas oportunidades para pular de um galho a outro, ou saltar para cultivar velhos caminhos cujas sequências foram interrompidas (Idem, p. 22, tradução nossa).

No contexto do estudo de caso dessa pesquisa, a inspiração trazida pelas autoras de que contextos entrelaçados aumentam a fluidez de caminhos e conferem maior oportunidade para a criatividade pode ser útil para o contexto das comunidades FLOSS. Uma comunidade FLOSS pode ser vista como um contexto institucional onde as regras e hierarquias não estão suficientemente claras e onde as decisões são tomadas em mais de um espaço, principalmente quando o grupo de desenvolvedores centrais faz parte de mais de uma organização. Nesse caso, ainda que uma comunidade FLOSS provavelmente não possa ser definida como um contexto entrelaçado, as consequências para a atuação da criatividade política podem ser bem parecidas. Olharemos para isso durante esse estudo. Sem mais delongas, passemos agora para a apresentação do caso.

3. Metodologia

[17] Nota de tradução: No texto original em inglês essa palavra aparece como "palate" no sentido de paladar, miríade de sabores. Optamos por utilizar o termo "leque" na tradução, já que é uma expressão mais comum da língua portuguesa que leva ao mesmo sentido almejado pelas autoras

Esta pesquisa está baseada em um estudo de caso de uma comunidade FLOSS, o Noosfero. Esse estudo de caso permitirá entender com mais profundidade como atuaram os agentes da comunidade, com os comportamentos destinados a aumentar sua autoridade na comunidade, além das táticas utilizadas tanto por líderes quantos por seus membros para assegurar ou influenciar nas decisões sobre o software. O caso da comunidade Noosfero foi escolhido por ser uma comunidade FLOSS Brasileira permitindo a essa pesquisa contribuir para um estudo de caso nacional. Além disso o fato da comunidade contar com a participação de organizações bastante diversas, como cooperativas privadas, universidades, movimentos sociais e o próprio Estado (através da empresa pública de tecnologia, o Serpro) e ao mesmo tempo, apesar de toda essa heterogeneidade, ter permanecido unida em torno do mesmo software durante mais de 11 anos de existência também foi um critério relevante para essa escolha. Essa característica aumenta as chances de conseguirmos encontrar casos concretos de agência voltados a negociações e resoluções de conflitos e tensões que serão úteis para o universo de análise da autoridade e influência no contexto da tomada de decisão. Em outras palavras, se as decisões tendem a ser unânimes e os momentos de conflito mais raros, menos diversidade será oferecida ao pesquisador. Também foi especialmente importante, como critério de escolha, a longevidade de mais de 11 anos da comunidade Noosfero e o fato de não ter se fragmentado ao longo desse tempo. Isso permitiu a comunidade se mantivesse presente no contexto de análise como um único caso. Por fim, também foi importante o fato da comunidade ter desenvolvido seu trabalho utilizando plataformas de codificação social, justamente as que têm sido apontadas pela literatura como espaços de colaboração aberta. Existe um risco na escolha da comunidade, que é o viés de seleção. A definição prévia desses critérios específicos para orientar a seleção da comunidade que será estudada pode levar o pesquisador a estudar um tipo muito específico de comunidade. A partir daí, as generalizações podem ser válidas apenas para essa comunidade ou no máximo para esse tipo específico. A esta pesquisa caberá, portanto, apresentar hipóteses para trabalhos futuros.

Essa pesquisa vai focar na análise de entrevistas com um grupo heterogêneo de agentes da comunidade. Para a definição do grupo serão utilizados os dados obtidos a partir da plataforma de codificação social utilizada pela comunidade. Ainda que o acesso aos códigos e à convivência nessas comunidades seja aberto, o poder de influenciar as decisões sobre a construção do software não está igualmente distribuído. Apenas um grupo específico de desenvolvedores tem permissão incluir alterações no software. Nas plataformas de

codificação social, os códigos são construídos através de atualizações, chamadas de "commits". As atualizações podem ser de dois tipos: melhorias no software, acrescentando um novo recurso ou correção de erros. O método de trabalho para construção dessas atualizações é baseado numa lógica descentralizada em que diversas pessoas estão trabalhando em paralelo, através da divisão do código em distintos ramos ("branches"). A lógica de ramos permite que um ou mais desenvolvedores trabalhem de forma distribuída, no seu próprio ramo, sem impactar os outros grupos. O processo de incorporação de uma atualização no tronco principal do código da comunidade é normalmente realizado pelo grupo de desenvolvedores mais influentes, os assim chamados "commiters". O momento em que se concretiza a aceitação de uma atualização pela comunidade é quando ela entra no ramo principal, chamado de ramo "master". Quando uma porção de código entra nesse ramo, significa que ela foi aceita pela comunidade. E poderá estar na próxima versão do software. Quem define o que sai em cada versão é o "release manager" ou gestor de versão. Esse papel é normalmente exercido pelos "commiters", tanto em grupo quanto indicando um deles para cumprir esse mandato durante um período determinado.

A Figura 3, indicada abaixo, apresenta em forma de diagrama o processo de atualização de código nas plataformas de codificação social. Nesse exemplo, há dois ramos onde estão sendo trabalhados novos recursos para o software ("ramo-A" e "ramo-B"). Na parte central da figura corre o ramo principal, o "master". Quando um desenvolvedor ou um grupo decide trabalhar num novo recurso para o software, o primeiro passo é copiar o código para um novo ramo. Essa ação é chamada de "fork" (seta em vermelho saindo do ramo principal). O trabalho corre de forma independente em cada ramo, no caso tanto o ramo A quanto o B não precisam se coordenar. As atualizações em qualquer um dos ramos (representadas pelos círculos numerados) são feitas de forma totalmente independentes, tanto entre elas quanto em relação ao ramo principal. O momento de coordenação acontece quando os desenvolvedores responsáveis por cada ramo submetem o código de volta para o ramo principal (os também chamados "pull requests"). Nesse momento é que entra a ação dos commiters, que é de aceitar - ou não - as atualizações submetidas pelos desenvolvedores que estavam trabalhando nos outros ramos. No diagrama, a segunda atualização submetida pelo ramo B não foi aceita (representada pelo círculo com um "x" vermelho). Uma vez incorporadas as atualizações com os novos recursos, entra o papel dos gestores de versão que podem optar por incorporar mais atualizações antes de publicar uma nova versão do software.

No diagrama, o ramo principal tinha a versão 0,1 como a mais atual. Após as atualizações incorporadas dos ramos A e B, o gestor de versão (“release manager”) decidiu que essas atualizações já eram suficientes para uma nova versão do software, nesse caso a 0,2 (as versões estão representadas pelos triângulos rosas numerados por 0.1 e 0.2).

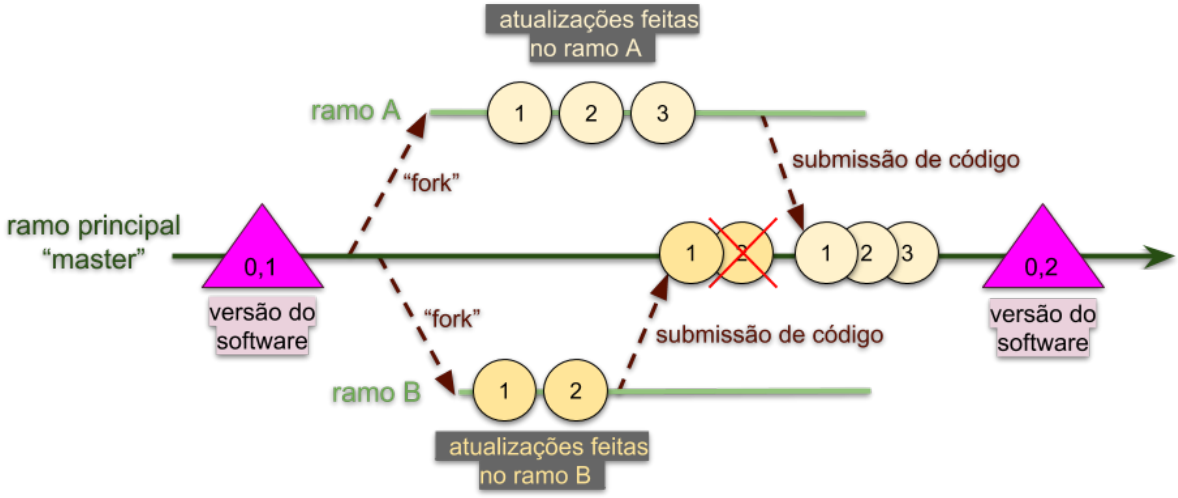

Figura 3. Incorporação das atualizações nas plataformas de codificação social

Fonte: Produção própria

O foco desta pesquisa será centrada na atuação dos agentes na tomada de decisão sobre incorporação de código, ou seja, o processo de incorporação dos “commits” no ramo principal do repositório de código da comunidade. Há códigos que entram simplesmente para corrigir pequenas falhas e outros que trazem novos recursos para o software. Há também momentos de grandes atualizações em várias partes do código, conhecidos no jargão técnico como “refatoramento”, ou seja, quando muitas partes do código são reescritas para fins de atualização da versão de alguma biblioteca[18] importante ou para melhorias de desempenho. Códigos mal escritos acabam rodando mais lento e esse tipo de melhoria é crucial para o bom desempenho de um software.

As decisões sobre as atualizações de código podem gerar grandes conflitos, principalmente quando trazem novos recursos. A decisão sobre novos recursos pode causar

[18] Os códigos de um projeto de software fazem referência e utilizam pedaços de código genéricos já escritos. Esses pedaços de código são chamados de bibliotecas e são normalmente mantidos e atualizados por grupos externos ao projeto/comunidade, muitas vezes outras comunidades. Quando essas bibliotecas são atualizadas para novas versões, os códigos do projeto que as utiliza precisam ser reescritos para continuar funcionando ou para fazer uso de recursos mais avançados.

uma mudança no status-quo da comunidade ou sacramentar a derrota de grupos com uma ideia distinta das prioridades. Essas decisões também revelam o componente político intrínseco a cada atualização. Independente do método decisório da comunidade, seja ele concentrado nos principais desenvolvedores ou realizado através de votação com o grupo mais amplo, essas decisões tendem a representar a vitória de algumas teses em detrimento de outras. São decisões, portanto, eminentemente políticas. Dessa forma, a pesquisa buscou estabelecer hipóteses de como os membros da comunidade influenciam o desenvolvimento de um FLOSS. Analisar e entender os comportamentos e a influência dos atores envolvidos permitirá que o fenômeno das comunidade FLOSS possa ser melhor compreendido.

As observações desta pesquisa foram obtidas de duas maneiras. Primeiramente foram analisados os dados referentes às transações nos repositórios de código para identificar os atores que participaram da construção dos códigos e alguns indicadores básicos sobre esse trabalho, como quantidade e tipo de contribuições (adição ou subtração de código) e concentração das contribuições no código. Em seguida, foram realizadas 15 entrevistas semi-estruturadas apenas com quem apareceu como contribuidor de código nos dados analisados do repositório.

Em suma, foram utilizadas as seguintes fontes de dados para análise:

A. Contribuições ao FLOSS Noosfero desde seu início (2007) até os dias atuais (incluindo o nome e email de cada contribuidor) / Fonte: Clonagem do repositório no Gitlab.com[19] / Método de coleta: Os dados foram gerados via linux shell a partir dos metadados do GIT presentes no repositório
B. 15 Entrevistas com pessoas membros da comunidade / Fonte: Gravação e transcrição / Método de coleta: Entrevistas semi-estruturadas

3.1. Dados do repositório

O protocolo GIT, adotado pela esmagadora maioria das plataformas de codificação social, mantém o registro de todas as atualizações realizadas no repositório, principalmente aquelas incorporadas no ramo principal. Para os outros ramos, o dado não é tão confiável, pois quando um commiter apaga um ramo antigo, todas as atualizações referentes a ele são

[19] O Gitlab.com é uma plataforma de codificação social alternativa ao Github.

removidas junto. Como o interesse da pesquisa foi justamente as atualizações operadas no ramo principal do software, ou seja, aquelas que foram de fato incorporadas pela comunidade, a eventual imprecisão desse tipo de dado não nos afetou.

Cada atualização (representada pelos círculos na figura 3 logo acima) contém informações sobre o autor, data, descrição e os arquivos de código que foram alterados pela atualização, indicando a quantidade de linhas que foram adicionadas e removidas. Na figura 4 abaixo está representada uma atualização, exatamente na forma com que geramos a lista de atualizações que foi consumida pelo software de análise dos dados.

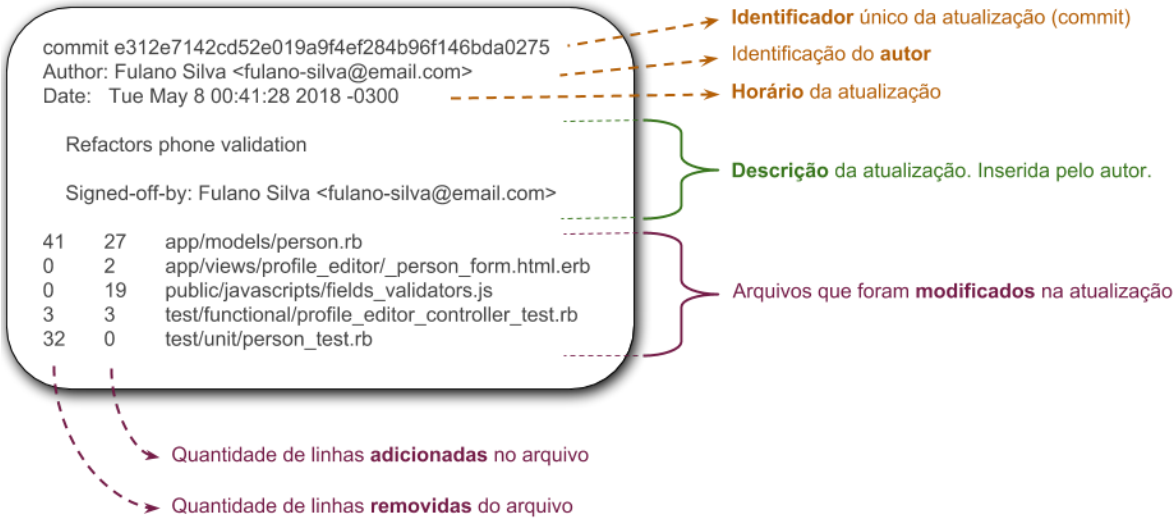

Figura 4. Representação de um commit registrado pelo protocolo GIT

Fonte: Produção própria

A partir desses dados, carregamos todas as atualizações da história do Noosfero no EaszyBI[20], um software de BI em nuvem que possui um filtro para ler esse formato de dados. O primeiro passo foi definir o elemento temporal para análise. Como os dados se referem a mais de 11 anos de contribuições no software (junho de 2007 a outubro de 2018) as análises utilizam o período semanal como o menor elemento temporal. O segundo passo foi eliminar das análises as alterações em partes do código com pouca relevância, como pastas de tradução, documentação e aquelas que continham software externo ao Noosfero. Essas pastas foram retiradas da análise por conta do nosso recorte ser focado nas alterações de código do

[20] Endereço para utilização do software EazyBI: https://eazybi.com

próprio Noosfero, que de alguma maneira acrescentem ou alterem funcionalidades existentes. Resoluções de falhas (bugs) também entram nesse recorte pois elas alteram as funcionalidades do software de forma que o conjunto todo passe a funcionar sem aquele problema específico. No Anexo A listamos quais pastas foram utilizadas na análise e as que foram removidas, com os seus respectivos motivos. O universo de contribuições analisadas se consolidou em 56.162 atualizações (commits) realizadas por aproximadamente 300 pessoas em 586 semanas (aproximadamente 11 anos e 4 meses).

Os dados foram utilizados para identificar o universo de membros da comunidade que seriam selecionados para a entrevista. Dos aproximadamente 300 membros que fizeram contribuições no código durante esse período, o nosso interesse foi naqueles que estavam ativos em momentos que se destacaram dos demais. A definição de momentos destacados foi obtida a partir de uma análise dos próprios dados, num processo ainda relativamente experimental. Como o escopo dessa pesquisa não permitiu um mergulho detalhado em cada momento, lançamos 3 hipóteses do que poderia ser considerado um momento destacado da comunidade Noosfero, sempre a partir de uma análise macro dos dados. Fizemos isso considerando a conjugação de três fatores distintos:

- FATOR MUITA COISA NOVA (Fig 5): Momentos em que as atualizações culminaram em muitas mudanças (adição ou subtração de código), com foco especial nas adições. Isso é relevante pois momentos de muita adição implica em criação de código novo, seja uma nova funcionalidade, seja uma grande refatoração;
- FATOR TENDÊNCIA DE DISPERSÃO (Fig. 6): Grande quantidade de atualizações (commits) com tendência de dispersão no código. A quantidade de atualizações não necessariamente segue a quantidade de adição ou subtração de código. Cada atualização pode fazer tanto poucas quanto muitas mudanças. Então é possível haver momentos em que muitas atualizações fizeram poucas mudanças nas linhas do código enquanto outros momentos podem haver poucas atualizações com muitas mudanças. Por isso a importância de considerar o elemento das atualizações. Uma das formas de saber se uma atualização fez parte de um momento destacado é tentar inferir o quanto ela aparece dispersa no código. Atualizações em várias partes do código tem maior probabilidade de serem responsáveis por grandes mudanças pois atualizações

pontuais tendem a estar mais concentradas, como resolução de bugs simples, por exemplo. A tendência de dispersão foi calculada considerando o quanto o número de atualizações aumenta quando o universo de todas as pastas do código (e não apenas as mais relevantes) é considerado, dentro da mesma semana. Quando essa tendência é mais próxima de "1" o aumento foi pequeno ou nulo, mostrando que incluir mais pastas na análise não altera a tendência. Esse indicador manifesta apenas uma tendência, pois é possível que as atualizações estejam dispersas dentre as pastas relevantes, e isso não é captado por esse número. Por isso essa tendência deve ser considerada com ressalvas;

- FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Momentos em que há muita adição de código no ramo principal e pouca diferença dele com os outros ramos, indicando que quase tudo foi incorporado. Outros momentos em que a quantidade de adição nos outros ramos é grande e no principal não aparecem grandes adições pode indicar que não houve incorporação do código realizado nos outros ramos. Enfrentamos algumas dificuldades na confiabilidade desse dado, já que parte do código dos outros ramos pode ter sido apagada do repositório por trabalhos de limpeza de ramos antigos. Isso faz com que não tenhamos mais acesso ao universo completo de todos os ramos. Por isso esse dado deve ser tratado com essas ressalvas.

A partir da conjugação desses fatores, foram produzidas três análises com o objetivo de pinçar os momentos que se destacavam. As figuras 5, 6 e 7 demonstram como essa seleção inicial foi feita.

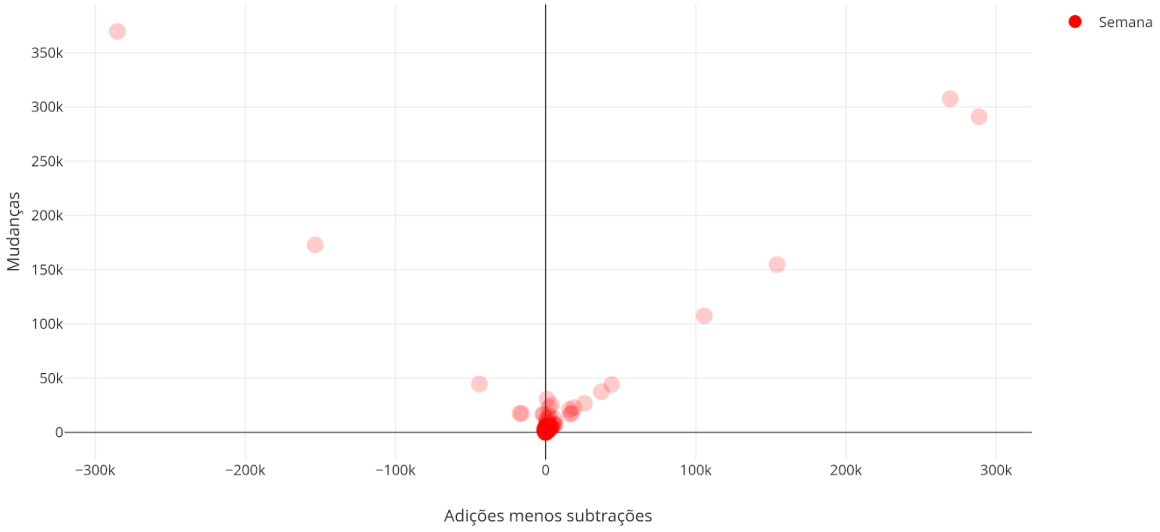

Figura 5. FATOR MUITA COISA NOVA: Destaques pela quantidade de mudanças e pelo alto saldo de adições

Fonte: Produção própria

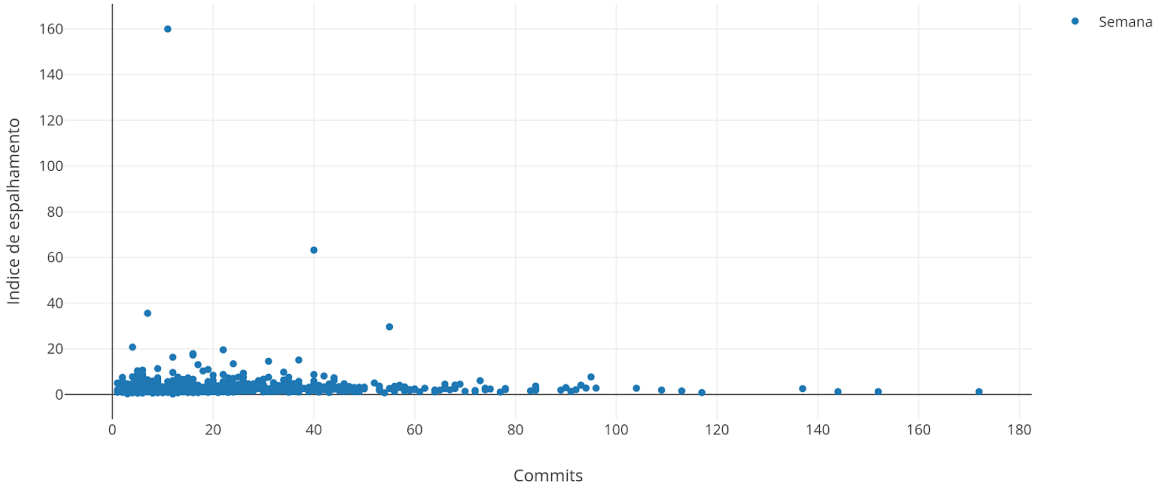

Figura 6. FATOR TENDÊNCIA DE DISPERSÃO: Destaque pela quantidade de atualizações (commits) e pelo índice de dispersão

Fonte: Produção própria

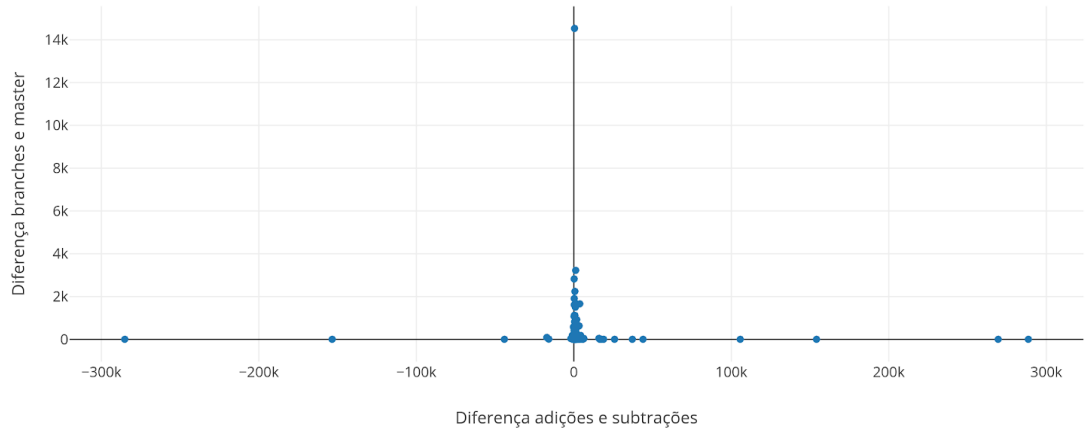

Figura 7. FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Destaque de muita adição e muita diferença entre o ramo principal e os outros

Fonte: Produção própria

Como fruto dessa análise foram pinçados 38 momentos. Desses, um novo filtro foi aplicado, dessa vez manualmente, selecionando momentos onde mais de um fator apontados pudesse estar presente. Com esse novo filtro chegou-se a lista final de 19 momentos destacados. No Anexo B apresentamos as tabelas que explicam como fizemos a filtragem dos 38 momentos para chegar na lista final de 19. Tratando os dados dos autores que aparecem nos momentos selecionados, foi possível comparar a quantidade de vezes que eles aparecem com a posição de cada um no ranking global de contribuições de toda a história do Noosfero até outubro de 2018.

Contribuidor	Vezes que aparece nos momentos chave	Posição no ranking global de contribuidores noosfero	Número global de contribuições
Rodrigo Souto	11	1	7.831
Antonio Terceiro	10	2	7.753
Braulio Bhavamitra	9	3	6.053
Daniela Feitosa	7	6	3.044
Larissa Reis	7	12	1.363
Leandro Nunes dos Santos	7	4	4.513
vfcosta	7	8	2.553
Aurélio A. Heckert	6	15	531

Joenio Costa	6	7	2.729
MoisesMachado	4	10	1.626
Tallys Martins	4	29	233
Arthur Del Esposte	3	26	270
Gabriela Navarro	3	37	126
Marcos Ronaldo	3	19	365
Rafael Reggiani Manzo	3	9	1.915
Caio SBA	2	14	568
Carlos Purificacao	2	35	152
Gabriel Silva	2	11	1.567
Luciano Prestes Cavacanti	2	65	43

Tabela 1: Posição dos desenvolvedores conforme participação nos momentos destacados e no ranking geral de contribuição para a comunidade

Fonte: Produção própria

Assim então, foi formada a lista inicial de pessoas que foram entrevistadas pela pesquisa. Dessa lista de 19 pessoas que aparecem pelo menos duas vezes nos momentos chave, foi possível entrevistar 13. Além dessas mais duas foram entrevistadas por terem sido mencionadas por dois entrevistados do primeiro grupo. Outro indicador interessante dessa lista é que 11 dos entrevistados está entre os 12 maiores contribuidores de noosfero. Isso apenas revela que os desenvolvedores ativos nos momentos destacados também foram os grandes contribuidores globais do noosfero. Isso ocorre porque a quantidade de contribuições globais de toda a história da comunidade é bem concentrada: das 56.162 contribuições do período estudado, 46.048 (ou seja, aproximadamente 80%) foram feitas pelos 15 maiores contribuidores.

3.2. Dados das entrevistas

Com base na identificação dos autores das contribuições nos momento destacados, foi formado o primeiro grupo de entrevistados. Desse grupo inicial, foi possível realizar a entrevista com pelo menos 13 pessoas e mais duas obtidas através de indicação dos próprios entrevistados. O roteiro foi produzido com base no referencial teórico estudado, buscando

identificar a agência dos atores que fossem relevantes para as categorias teóricas dos comportamentos para progressão da autoridade lateral na comunidade e as táticas do membros e líderes para influenciar no processo de decisão. O roteiro integral se encontra no Anexo D.

As entrevistas foram realizadas utilizando o aplicativo google hangouts, com exceção de uma que foi presencial. Todas foram gravadas e transcritas. O trabalho de coding foi realizado utilizando um CAQDAS, "Computer-assisted qualitative data analysis software" chamado RQDA[21], que é um pacote baseado em R utilizado para análise qualitativa de dados. O trabalho de coding se baseou na mistura de duas técnicas o "structural coding" e o "descriptive coding". Sobre o primeiro, SALDAÑA (2009, p. 66) explica que,

> Structural Coding is a question-based code that "acts as a labeling and indexing device, allowing researchers to quickly access data likely to be relevant to a particular analysis from a larger data set" (Namey, Guest,Thairu, & Johnson, 2008, p. 141 apud SALDAÑA, 2009, p. 66). Structural Coding both codes and initially categorizes the data corpus.The Sources suggest that Structural Coding is perhaps more suitable for interview transcripts than other data such as researcher-generated field notes, but open-ended survey responses are also appropriate with this method.

No nosso caso, a técnica de "structural coding" foi usada com base nas categorias de autoridade lateral, tática dos membros e líderes, tipos de líderes e tipos de membros. A figura 8 abaixo exibe um diagrama dos códigos que foram utilizados nesse método. A maior parte deles foi criada antes do início das análises mas algumas novas surgiram a partir da leitura das entrevistas com as pessoas.

[21] Link de acesso ao CAQDAS RQDA utilizado na pesquisa: http://rqda.r-forge.r-project.org/

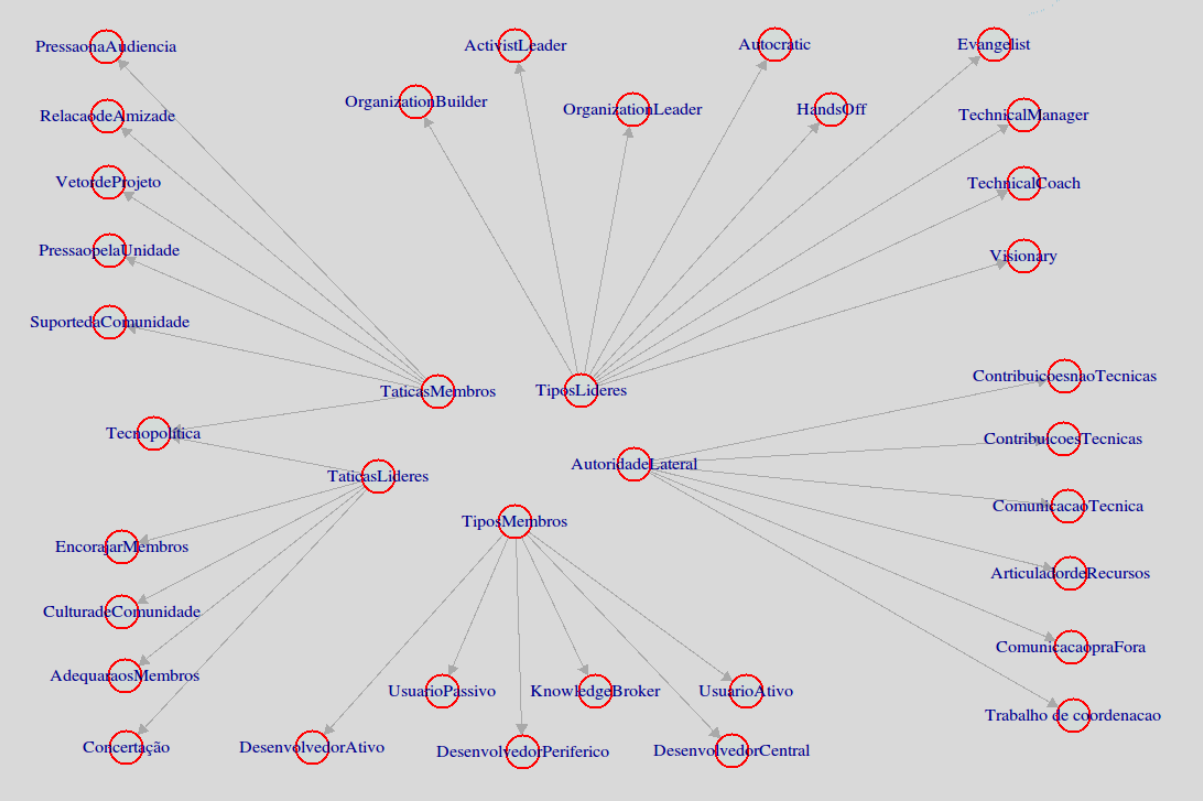

Figura 8. Códigos utilizados na técnica “structural coding”

Fonte: Produção própria a partir da exportação do software RQDA

Em paralelo também utilizamos a técnica do “descriptive coding”, que consiste em identificar na leitura das entrevistas, tópicos importantes e ao mesmo tempo relevantes para a pesquisa. Tesch (1990, p. 119 apud SALDAÑA, 2009, p. 70) explica que a técnica

> Descriptive Coding summarizes in a word or short phrase – most often as a noun – the basic topic of a passage of qualitative data. To clarify, Tesch (1990) differentiates that “it is important that these [codes] are identifications of the topic, not abbreviations of the content. The topic is what is talked or written about.The content is the substance of the message”.

No Anexo C apresentamos uma tabela onde é possível ver a lista de códigos criadas a partir da utilização da técnica “Descriptive Coding” com uma breve explicação de seu significado. Foram códigos criados com base nos assuntos que surgiram da leitura das entrevistas e que ao mesmo tempo possuíam relação com as perguntas colocadas pela pesquisa. Alguns códigos que criados na primeira rodada de codificação, foram posteriormente unidos com códigos mais relevantes, totalizando uma lista final de 21 códigos obtidos através desse método.

4. Apresentação do caso

4.1. A façanha do Noosfero

A comunidade Noosfero nasceu de um projeto comercial da Colivre, uma cooperativa de software livre da Bahia, que em 2007 vislumbrou a oportunidade de criar uma plataforma livre que atendesse a demanda de dois clientes, sendo um deles uma rede vinculada ao movimento social de economia solidária no Brasil: "O principal é isso, dois clientes, aparentemente com necessidades diferentes, que a gente percebeu que poderia ser um software só que incluísse as duas necessidades" (Entrevistado 2, Commiter/RM, Colivre, [593:934]). Contar a história do Noosfero somente a partir da lógica contratual da Cooperativa Baiana com seus clientes é perder um contexto importante da história que ainda é bastante vivo na memória de que participou desse processo. Há dois aspectos importantes dessa história que evidenciam a relação seminal do Noosfero com os movimentos sociais do Software Livre e da Economia Solidária.

O primeiro aspecto diz respeito à enxergar a Cooperativa e a criação do Noosfero como algo que buscava influenciar a sociedade de acordo com os valores dos dois movimentos. Isso fica evidente na associação que os membros da cooperativa fazem dela com os interesses mais gerais do movimento software livre:

> Ter instituições como a colivre no mundo, na minha concepção ideológica de mundo é importante, instituições, empresas que pensem em bem estar social, em ser remunerado de uma forma justa mas também estando num contexto onde os outros podem ganhar, um contexto de ganha-ganha. Software livre sempre fez parte da minha formação, desde que eu entrei na faculdade eu já entrei adorando a ideia do software livre, então isso sempre veio comigo também, logicamente se o software fosse proprietário eu também não teria esse tipo de relação porque eu acredito tb nessa ideia de que o conhecimento deve ser compartilhado, então é um motivador, ser software livre ser uma organização desse estilo e a relação de amizade com as pessoas que fazem parte do projeto porque um alimenta a outra de certa forma (Entrevistado 15, Commiter, Serpro, [5750:6539]).

E também em como a parceria com o Movimento Social de Economia Solidária foi uma decisão tomada em função dos objetivos do ativismo político:

> [A gente discutia muita política], até por conta do nascimento do noosfero casado ali com o Fórum Brasileiro de Economia Solidária né, então era o meio, a ferramenta para potencializar economicamente os empreendimentos de economia solidária no brasil, no modelo de economia solidária também. ou seja, um meio pra fomentar e deslanchar os empreendimentos, usando uma lógica de economia solidária também, o FBES [Fórum Brasileiro de Economia Solidária] poderia contratar a Microsoft pra resolver o problema dela se tivesse dinheiro pra isso, mas não resolveu contratar uma cooperativa resolveu fazer nesse modelo (Entrevistado 1, Commiter/RM, Colivre, [45504:46280]).

O segundo aspecto da evidente associação da criação da comunidade com os objetivos dos movimentos sociais diz respeito às motivações pessoais para participar do projeto. Percebemos que a fundação do Noosfero foi uma realização de desenvolvedores com perfil de ativismo:

> Meu interesse no Noosfero é paralelo, igual ao meu interesse de ter entrado nesse empreendimento colivre, que é contribuir com projetos de software livre, trabalhar com projetos de software livre, que era algo que eu já fazia anteriormente na militância, no ativismo, já participava da comunidade local de software livre da bahia, e meu interesse era nesse sentido, trabalhar com SL [Software Livre], trabalhar profissionalmente com isso. Fomentar o software livre no Brasil e no mundo, não engrossar mais ainda o caldo dos softwares proprietários, desse modelo de conhecimento fechado, proprietário (Entrevistado 1, Commiter/RM, Colivre, [1924:2492]).

A ponto da participação na comunidade ser vista tanto como uma trajetória de aprendizado "em relação ao software livre, em relação a algumas filosofias desenvolvidas ali nesse meio" (Entrevistado 6, Desenvolvedor comum, UnB, [51437:52276]), quanto como uma prova de alinhamento a certos princípios, para além da aptidão técnica/profissional:

> Eu tenho uma visão assim, ninguém permanece muito na comunidade software livre, por ser uma coisa voluntária que demanda vários valores, éticos e pessoais então ninguém tá muito tempo nisso se não compartilhar certos valores (Entrevistado 3, Desenvolvedor comum/Articulador de recursos, USP e UnB, [28357:29417]).

Chegando ao ponto da comunidade, quando já formada e ativa, ser comparada explicitamente com um movimento social:

> Eu imagino a comunidade noosfero como um movimento social na verdade, uma comunidade de software livre de uma forma mais geral eu imagino muito parecido com a dinâmica que acontece dentro dos movimentos sociais (Entrevistado 10, Desenvolvedor comum, EITA, [21722:23128]).

Naquele momento, os desenvolvedores do Noosfero eram todos vinculados à cooperativa, seja como cooperados ou como estagiários, e não havia uma comunidade formada. A comunidade começa a se formar quando surge um braço de desenvolvimento, vinculado ao movimento de economia solidária, que assume parte do desenvolvimento que antes estava a cargo apenas da Colivre. Esse braço foi a EITA (Cooperativa de Trabalho Educação, Informação e Tecnologia para Autogestão), uma cooperativa voltada a oferecer tecnologias para os empreendimentos de economia solidária. A EITA era responsável por fazer a gestão do portal Cirandas, que, baseado em Noosfero, era oferecido como serviço para os empreendimentos de economia solidária. Com força de desenvolvimento própria, a EITA foi responsável por desenvolver diversos pedaços do software Noosfero, principalmente aqueles voltados para atender as necessidades do Movimento de Economia Solidária. O objetivo era criar uma rede social que ajudasse a dinamizar a circulação de produtos entre os empreendimentos, coletivos e fornecedores:

> A gente também foi muito ambicioso e pensou num sistema que trabalhasse em rede, que conseguisse ver a trajetória de comercialização de um produto em rede. Então, a gente queria que fosse possível a gente vender um produto de um fornecedor pra um coletivo. E esse coletivo vender pra outro coletivo ou outra loja. Ou um fornecedor vender pra um empreendimento que vende pra um coletivo, ou seja, uma rede de "n" compradores e fornecedores né (Entrevistado 5, Commiter, EITA, [5771:6533]).

Nesse momento ocorre a primeira expansão da comunidade de desenvolvedores Noosfero e a partir disso a comunidade passa a ser constituída por duas organizações atuantes no desenvolvimento do software, a Colivre e a EITA. A comunidade volta a crescer com a chegada das universidades. A primeira a entrar é a Universidade de São Paulo (USP) quando começa a utilizar o Noosfero como base da rede social de alunos, o STOA Social[22]. Quem assume o desenvolvimento da rede é a Colivre, porém como o Noosfero vira objeto de uma disciplina do curso de computação da USP, alguns alunos começam a se envolver no desenvolvimento da plataforma, constituindo assim a segunda expansão da comunidade de desenvolvedores. Uma nova expansão da comunidade acontece quando a Faculdade do Gama, da Universidade de Brasília (UnB), mais especificamente o Laboratório Lappis[23], adota o Noosfero como a tecnologia base do portal da Faculdade, em paralelo à adoção dele em

[22] O Stoa Social ainda pode ser acessado nesse link: https://social.stoa.usp.br/

[23] O laboratório Lappis é uma estrutura ligada a Faculdade do Gama (UnB) ao curso de Engenharia de Software e é gerido por um grupo de professores do curso. Tem histórico de trabalho com software livre e metodologias ágeis

algumas disciplinas do curso, para logo em seguida adotá-lo como objeto de uma parceria com o Ministério do Planejamento[24]. A entrada da UnB na comunidade, a partir dessas três frentes (Portal da FGA, Disciplinas e projeto com o Ministério do Planejamento) traz um aporte considerável de recursos financeiros e não financeiros, dinamizando bastante a comunidade. Em paralelo a isso, já avançava o processo de adoção do Noosfero pelo Serpro, para posteriormente ser adotado como tecnologia para algumas das iniciativas de e-participação da Presidência da República. A entrada do Serpro, que chegou a ter um time de 5 desenvolvedores trabalhando tempo integral em Noosfero, também foi um momento bastante dinâmico para a comunidade. Na figura 9 é possível ver uma representação gráfica da linha do tempo de entrada das organizações na comunidade.

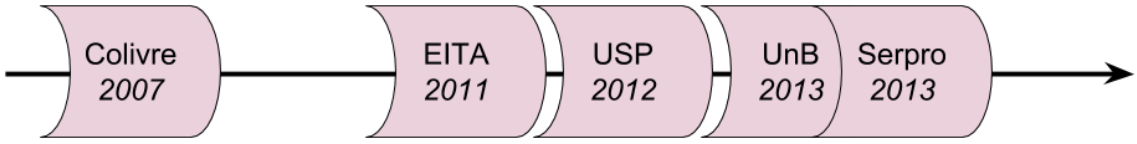

Figura 9. Linha do tempo que representa o crescimento da comunidade Noosfero

Fonte: Produção própria

A partir dessa leitura, foi possível perceber que a comunidade do Noosfero teve um crescimento em saltos, cada salto representando a entrada de uma nova organização com força de desenvolvimento. Novos desenvolvedores chegavam sempre que uma organização adotava o software e se organizava para desenvolvê-lo. Nesse sentido a plataforma de codificação social servia como base para a atuação dessas organizações, sendo mais raro o caso de indivíduos que isoladamente se interessassem e começassem a contribuir com o software:

> Não é uma comunidade imensa a comunidade noosfero, e a maioria das pessoas que tá trabalhando com noosfero tá envolvida em algum projeto e de alguma forma está sendo remunerado. São poucas pessoas que participam da comunidade assim de forma voluntária mesmo (Entrevistado 1, Commiter/RM, Colivre, [13325:14319]).

[24] O laboratório Lappis já contribuia com o Noosfero através de instrumentos próprios do curso de engenharia de software da UnB, porém quando é efetivada a parceria para o desenvolvimento da nova versão do Portal do Software Público Brasileiro com o Ministério do Planejamento essa contribuição se intensifica

A história do Noosfero é uma história de organizações e projetos em torno de um software, cada uma utilizando-o para seus objetivos específicos. É a história de um software que serviu de base para iniciativas tão díspares quanto um portal para empreendimentos de economia solidária, passando por rede social de blogs e de alunos em universidades, até servir de base para iniciativas de e-participação.

Dentro da tipologia oferecida por NAKAKOJI, YAMAMOTO & NISHINAKA (2002), o Noosfero é um FLOSS orientado a serviço que, segundo os autores, tende a desenvolver mecanismos mais sofisticados de governança. Como veremos a seguir, o Noosfero se destaca nesse sentido por oferecer principalmente dois serviços: publicação de conteúdo e rede social. Esses serviços são bastante genéricos, podem ser usados para uma infinidade de objetivos. Na prática, as organizações recombinavam os recursos genéricos do Noosfero de diversas formas, com objetivo de atender suas diferentes necessidades, todos desenvolvendo dentro do mesmo software:

> O Noosfero sempre foi muito ambicioso, muito diversificado. o Noosfero na verdade são vários produtos num software só né. Ele é comércio, ele é rede social, ele é blog, ele faz jornalismo, ele é muita coisa completamente diferente. Então, nunca teve né um objetivo comum. O objetivo comum é vamos compartilhar isso daqui, vamos juntar esforços pra que uma coisa que você fez eu possa reutilizar né. A gente tem um software nacional, um software produzido aqui no Brasil, um software livre que seja bom e tal. Mas objetivo mesmo na área de produto né, no final você ta desenvolvendo um produto né que vai ser usado por um usuário, isso nunca teve porque tem uma diversidade muito grande. Então, cada organização tinha o seu objetivo (Entrevistado 5, Commiter, EITA, [15071:16033]).

Essa característica era algo marcante do Noosfero, que tinha que atender muitas necessidades, dentro da mesma comunidade:

> Eu acredito que tem a ver com algo específico do Noosfero sim, por ele ser um projeto que tenta ser muita coisa né, tenta ser muito flexível, tenta ser matricial, que já é uma coisa complexa, uma coisa grande né, tenta gerar muitos conteúdos diferentes, tenta ter vários tipos de perfis, etc. Eu acho que é uma coisa de produto mesmo do Noosfero sabe, quer queira quer não ele vai crescendo muito organicamente assim, sabe. Cada um vai fazendo o que pode, o que quer dentro dele, vai entrando assim, desde que seja uma contribuição razoável, mas eventualmente vai virando uma coisa muito grande e complexa, difícil de manter (Entrevistado 6, Desenvolvedor comum, UnB, [46419:47073]).

Do ponto de vista do software, essa característica pode ser vista como um defeito. Um defeito porque um software que tenta fazer muita coisa, corre o risco de não fazer nada bem. E para alguns, isso aconteceu com o Noosfero:

> Todo mundo queria usar e cada um queria usar de um jeito. E era ruim isso, de toda hora a gente ter que deixar um monte de opções disponível né. Então, a gente disputava isso bastante. Mas eram essas as discussões, a gente tinha vinha muito disso: "Ah, mas é melhor a gente deixar o usuário forçar dessa forma? Não, a gente tem que deixar opcional. A gente é uma comunidade. Tal projeto que não pode usar isso nunca, que não pode aparecer isso." Esse era o grande… que eu acho que acabou gerando esse defeito técnico do Noosfero de ter coisas demais né, fazer coisas demais (Entrevistado 6, Desenvolvedor comum, UnB, [28160:29646]).

Por outro lado, as discussões sobre a qualidade do software não nos interessam aqui. O que queremos observar é que, em termos de uma comunidade formada por pessoas vinculadas a diversas organizações, esse foi um dos principais méritos do Noosfero, ou seja, ter conseguido manter durante tantos anos uma comunidade unida em torno de um software que atendesse tantas necessidades diferentes. Essa foi a grande façanha do Noosfero, e conseguiu fazer isso desenvolvendo mecanismos para garantir a autonomia dos grupos de desenvolvedores, ao mesmo tempo preservando a interdependência das suas contribuições:

> Grupos que tem interesses específicos e boa parte desses interesses são intersecção com o projeto noosfero acabam se interagindo ali pra ver a melhor forma com que esses interesses sejam juntados pra formação do projeto, mas eu não acho que exista um objetivo da comunidade como um todo do que quer ser de futuro. Existe até de certa forma tá até se tentando buscar isso mas, não é uma coisa que é planejada, cada grupo que está interagindo, em momentos distintos foi o Serpro um grupo, eu outro momento UnB foi um grupo, em um momento a USP foi um grupo, que tinha um interesse, a Colivre sempre foi um grupo que tinha um interesse que tava constante ao longo de todo o tempo porque ela sobrevive daquele software também, esses outros grupos vinham, interagiam, alguns ficavam outros saiam (Entrevistado 15, Commiter, Serpro, [6624:8477]).

Para entender a relação entre as organizações, é fundamental conhecer como funciona a governança da comunidade Noosfero. A progressão dos atores da periferia em direção ao centro da comunidade é realizada através dos papeis de desenvolvedor comum, commiter e RM ("release manager"). Qualquer pessoa pode ser desenvolvedor comum, bastando submeter contribuições. A diferença entre qualquer desenvolvedor e um commiter, é que o segundo tem permissão para escrever código no ramo principal da comunidade. Se fossemos comparar com a lógica de colaboração num documento online, o commiter pode editar o

documento (remover e acrescentar texto) enquanto o desenvolvedor comum só pode ler e comentar. Em outras palavras, os commiter são os DONOS do documento, enquanto os outros desenvolvedores são COLABORADORES que submetem sugestões de mudanças. Dessa forma ser commiter é uma questão de poder, acompanhada de um senso de responsabilidade:

> [Ser commiter é] poder escrever na raiz principal, ter poder. Eu acho que o principal é aquela coisa que você se torna mais responsável, porque queria ou não, se você escreve na raiz principal você tem que saber que tudo que você coloca lá vai ter uma responsabilidade por trás daquilo" (Entrevistado 2, Commiter/RM, Colivre, [21681:23261]).

A questão do poder com responsabilidade também aparece numa descrição comparando as vantagens com as responsabilidades do "cargo" de commiter do noosfero:

> Tem vantagens é claro, mas é mais responsabilidade do que vantagem, na prática. Porque você como commiter acaba sendo responsável por revisar código pras pessoas, [e mesmo sendo commiter] qualquer código que você precise mandar precisa ser revisado por outro commiter também, salvo exceções no caso de ser muito simples ou no caso de ser uma coisa muito emergencial (Entrevistado 4, Commiter/RM, Colivre, [26294:27667])

Isso acontece porque existe uma política de desenvolvimento que limita o poder dos commiters. Embora eles tenham a possibilidade técnica (via permissões da plataforma de codificação social) de gravar diretamente na raiz, é um poder que não pode ser exercido sem os custos sociais de desrespeitar uma política de desenvolvimento acordada entre todos os commiters. Então de certa forma, o poder de gravar código diretamente é limitado por essas regras - que são externas à plataforma de codificação social - e a responsabilidade é maior pois o código escrito por um commiter fica sob a mira dos holofotes do próprio grupo de commiters, mais até do que o código escrito por um desenvolvedor comum.

Para um desenvolvedor comum se tornar commiter ele precisa atender a dois requisitos. O primeiro é demonstrar experiência com o código do noosfero através de um critério objetivo de quantidade de contribuições aceitas. O segundo requisito é que ele também precisa ser endossado pelos pares, ou seja, por pelo menos dois outros commiters. No processo de socialização da comunidade, ser commiter aumenta as chances do seu argumento ser escutado. Um argumento apresentado por um commiter é um argumento mais forte pois vem de alguém "está ativo, trabalhou bastante, tá por dentro da situação, então ele entende do código, então talvez acho que seus argumentos pra quando você quer mudar alguma coisa tenham mais força" (Entrevistado 7, Commiter, UnB, [18523:19584]).

O RM faz parte do grupo de commiters e tem a responsabilidade de publicar as versões do software. A decisão sobre incorporação de código é tomada pelos commiters individualmente ou em grupo, não cabendo ao RM individualmente. O RM só tem prevalência aos commiters em dois casos. O primeiro caso é quando ocorre algum conflito grave entre os commiters, que não foi resolvido de forma consensual. Nesses casos - que embora façam parte das regras da comunidade noosfero, nunca aconteceram na prática - a decisão final sobre incorporação de código é do RM. O segundo caso é quando uma incorporação vai prejudicar algum caso de uso ou organização que utiliza o Noosfero. É papel do RM portanto, garantir que a nova versão não quebre nenhuma instalação existente de Noosfero. Ainda que os commiters também compartilhem essa preocupação e recusem código por conta desse problema, a responsabilidade final nesse caso é do RM, pois é ele que tem a prerrogativa operacional de publicar a versão. Por conta disso ele pode até ser visto pelos commiters como uma liderança, mas seu poder é mais reativo. O poder do RM é reativo porque ele não pode se recusar a incluir uma atualização na versão se ela já tiver sido incorporada pelos commiters e não quebrar nenhum uso atual do Noosfero.

Salvo nesses dois casos, o poder do RM é o mesmo dos commiters. Sua distinção diz respeito a um trabalho organizativo, que é o de publicar e comunicar as versões. O principal núcleo decisório é o núcleo de commiters. Ainda que o RM possa ser visto como ocupando uma posição superior à um commiter, isso é só aparente. Sua prevalência é apenas operacional e para isso ele precisa coordenar o fluxo de informações nos momentos que antecedem a publicação das versões:

> Na prática o que entra na versão é o que já entrou, é o que já foi incorporado até o dia em que a versão vai ser lançada. No máximo, o que o release manager (RM) tem de poder, pra poder dizer que uma coisa entra ou não entra, é se por exemplo, se uma coisa tá no meio do caminho pra ser finalizada, ainda não terminou pra ser incorporada, aí a pessoa que tá desenvolvendo pede ao RM pra adiar um pouco o lançamento porque ela quer que aquilo entre nessa versão logo pra não ter que esperar muito pra entrar depois. Aí o release manager tem esse poder de decisão nesse momento e fala assim: “Não, eu vou seguir o cronograma e [...] se você conseguir terminar a tempo ela entra, se você não conseguir ela vai ficar pra próxima”. Ou então, o release manager fala assim: “Não, beleza, eu vou adiar um pouquinho até tal data e se você conseguir fazer, entra”. Então, é só esse nível de poder que ele tem na prática pra decidir se entra ou não entra (Entrevistado 4, Commiter/RM, Colivre, [61304:62663]).

O RM precisa estar sempre informado pois é ele que dá consequência para o que foi incorporado. Porém, as decisões são tomadas no processo contínuo de incorporação, ou seja, pelos commiters:

> Na minha opinião quem acaba tomando as decisões mais duras são assim os contribuidores mais ativos né. Assim, eu vejo muito como uma hierarquia mesmo né, então o papel do release manager é ouvir os desenvolvedores mais ativos e os desenvolvedores mais ativos estão em contato com a comunidade em si né. [...] Então, eu vejo que há uma…hierarquia não é a melhor palavra, mas tem um encadeamento pra todo mundo fazer a sua opinião chegar até o release manager né. Então, eu diria assim que o papel mais forte mesmo é dos desenvolvedores mais ativos né (Entrevistado 8, Desenvolvedor comum, USP, [4776: 5488]).

A figura 10 traz um diagrama ilustrativo do grau de influência de cada um desses três papéis na comunidade. A posição no círculo é apenas ilustrativa. Quanto mais próximo do centro, mais poder de decisão. Note que os desenvolvedores comuns ficam localizados na área de contribuição/sugestão, ou seja, eles só podem sugerir novos códigos já que a decisão de incorporação é dos commiters, localizados na área de decisão/incorporação. O RM também está localizado nessa área, mas não necessariamente é a figura com o maior grau de influência na comunidade, já que seu trabalho é apenas dar consequência, na publicação da release, nas decisões de incorporação que já foram tomadas pelos commiter ao longo do processo de desenvolvimento (revisão e incorporação, com ou sem discussão).

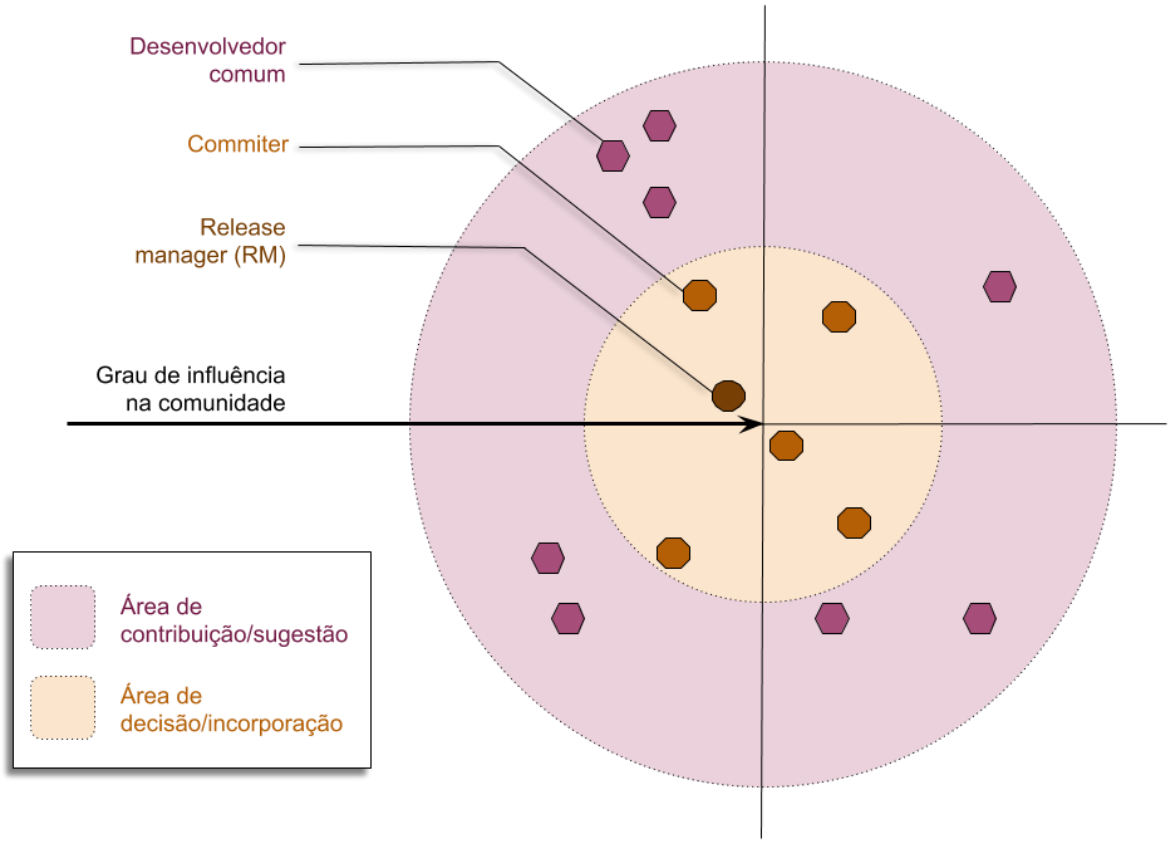

Figura 10. Gráfico de influência na comunidade de acordo com os 3 papéis

Fonte: Produção própria

As regras de governança da comunidade Noosfero não estavam prontas desde o início. Ao contrário, foram se aprimorando ao longo do tempo, obtendo o efeito de resolver - ou amenizar - conflitos e com isso manter a unidade social e de código da comunidade ao longo de mais de 11 anos. Diante disso, identificamos pelo menos três fatores que contribuíram para que a comunidade Noosfero tenha conseguido realizar a façanha de permanecer unida. O primeiro foi a permanência da Colivre como espinha dorsal da comunidade, uma espécie de "backbone organization", que garantia a continuidade do software nos momentos de crise, ao mesmo tempo que mediava os interesses da comunidade para evitar a divisão. O segundo fator foi a comunidade ter conseguido aprimorar os mecanismos de governança permitindo que as organizações conseguissem se planejar de forma autônoma, estabelecendo um processo transparente para os desenvolvedores periféricos se tornarem desenvolvedores centrais e com isso simplificando os passos para incorporação de código no software. Com isso, cada uma das organizações conseguiu promover pelo menos um de seus desenvolvedores como

commiter do Noosfero (algumas mais de um como UnB e Serpro). E finalmente, o terceiro fator foi a capacidade de construir uma arquitetura tecnopolítica no software que garantiu aos desenvolvedores salvaguardas para sua autonomia que foram muito utilizadas nos momentos de crise de governança. Apresentaremos a seguir cada um desses fatores mais detalhadamente.

4.1.1. Fator uma organização central (“backbone organization”)

A identidade da Colivre muitas vezes se confunde com a identidade do Noosfero. Apesar de ter sido criada antes do software, rapidamente o Noosfero se tornou o principal carro chefe da cooperativa. Essa coincidência de trajetórias se explica pelo fato da cooperativa ter criado o Noosfero como um projeto livre comercial e já logo nos primeiros anos ter ancorado sua sustentabilidade financeira em projetos baseados no software. O Noosfero não foi, inicialmente, criado para uma comunidade: “O noosfero começou pra atender demanda de clientes que chegaram na colivre, não foi um projeto que começou de forma a atender a comunidade, é um projeto comercial, o noosfero é um projeto livre comercial” (Entrevistado 1, Commiter/RM, Colivre, [14664:15206]). Durante a evolução do software e o crescimento da comunidade, a Colivre nunca deixou de ser a organização central do arranjo, mesmo quando a maior parte dos recursos investidos no software não provinha dos projetos sob sua coordenação direta. Em outras palavras, o Noosfero era de longe o principal projeto da cooperativa e qualquer distúrbio nele se refletiria quase que diretamente no seu ganha pão.

Por outro lado, apesar de nascer como um projeto livre comercial, o Noosfero nasce também como um Comum, ou seja, já nasceu licenciado como um recurso que, desde sua primeira linha de código, já podia ser estudado e apropriado por outros atores e estava submetido a certas regras de governança. O Noosfero, ainda que sem comunidade formada, já nasceu como um FLOSS. A filiação da colivre aos princípios do movimento Software Livre permitiu que a comunidade tivesse condição de surgir. A existência de um FLOSS não é condição suficiente para o surgimento da comunidade, mas é uma condição necessária. Sem acesso ao comum, a comunidade não tem como existir, pois não haveria um ativo em torno do qual os agentes colaborar. Se fôssemos fazer uma comparação, o Comum é a semente. Os nutrientes da terra quem dá é a governança da comunidade. No caso do Noosfero, foi a cooperativa quem forneceu os primeiros nutrientes para o germinar da semente, a partir de

uma postura apreciativa com os diversos objetivos de uso que as outras organizações tinham para o Noosfero:

> Eu acho que cada grupo tinha uns objetivos e principalmente a Colivre tentava fazer com que todos os atores fossem beneficiados assim, tentava conciliar, era um dos trabalhos do RM [“release manager, ou gestor de versão] inclusive, e de quem revisava código em geral, era perceber se tudo que entrava seria, poderia ser adaptado para todo mundo, cada funcionalidade, cada código que entrava, a gente tinha que entrar pensando nos objetivos que a gente achava, os objetivos que eram claros (Entrevistado 2, Commiter/RM, Colivre, [7893:8864]).

A história da comunidade Noosfero é uma história de organizações que, ao longo do tempo, vão aderindo ao desenvolvimento ativo desse comum. A primeira organização que faz esse movimento, a EITA, faz isso dentro da organização do Fórum Brasileiro de Economia Solidária, que foi um dos primeiros clientes a demandar a Colivre pela solução de software do Noosfero. O movimento que ocorre dentro do FBES foi então, após uma primeira etapa de desenvolvimento pela Colivre, do “cliente” se tornar “sócio” do projeto, assumindo uma parte ativa no desenvolvimento do software. Isso seria impensável num arranjo tradicional de cliente/fornecedor. Claro que essa não é uma história idílica do desenvolvimento harmônico e consensual de uma comunidade de ativistas desenvolvendo software. A história real é bem mais conturbada e cheia de ambivalências. Na história real a Colivre chegou a ser vista como uma organização centralizadora com um imenso temor em descentralizar o controle da comunidade e com isso colocar em risco a sua sobrevivência:

> Na época o Noosfero era muito fechado. Ele tinha muita dificuldade de dialogar com contribuições externas, praticamente nada fora da Colivre entrava no código do Noosfero. Eu fui uma das primeiras pessoas que tava cobrando que isso fosse possível acontecer na prática. Então, eu tive muito choque com a comunidade pra poder abrir ela e fazer com que ela se tornasse mais flexível e não, às vezes por se mostrar muito pedante, muito rígida, com um critério de qualidade muito alto que ninguém conseguiria contribuir código né (Entrevistado 5, Commiter, EITA, [6535:8004]).

Na história real, as organizações que aderem à comunidade o fazem muito mais preocupadas em resolver seus objetivos específicos do que em colaborar com a construção de um comum que atenda todos os casos de uso:

> Teve bastante gente que foi ativa, mas que num, sei lá, mas que não colocava essa responsabilidade de, por exemplo, se importar em fazer o lançamento do Noosfero que seria usado por todo mundo. Bastante gente que não era da Colivre, mas que agora é bem ativa, acabava que se preocupava mais com o seu, a sua plataforma, a sua instância e meio que passava a largo desse

> processo de release aí oficial, digamos assim. Enfim, a Colivre foi quem se propôs aí pra esse processo né de fazer o Noosfero ser um projeto de software livre e de fato disponibilizar aí pra comunidade (Entrevistado 11, Commiter/RM, Colivre, [36042:36732])

Na história real há espaço para todas essas contradições. Mas o resultado factual foi uma comunidade com organizações que faziam um uso bastante plural do software, que tinha tudo pra se dividir e não se dividiu. O resultado factual foi uma comunidade que amadureceu seu processo de governança mantendo um FLOSS brasileiro que dura até os dias de hoje. E isso se deve, em partes e com todas as suas contradições, ao papel desempenhado pela organização que criou o software e nutriu a comunidade nos seus primórdios. Essa foi portanto, a contribuição da Colivre para a façanha do Noosfero a que nos referimos agora pouco.

4.1.2. Fator protagonismo da comunidade

Como já vimos, a comunidade Noosfero surge no momento que organizações externas à Colivre começam a investir força de desenvolvimento na construção do software. Durante os primeiros anos, as decisões sobre o software permaneceram bastante centralizadas na Colivre, por questões que já tratamos aqui. A busca das organizações por mais autonomia nas decisões sobre incorporação de código somada ao acúmulo de trabalho de revisão a cargo da cooperativa, fez com que a comunidade Noosfero começasse a se abrir. O momento da abertura acontece também num momento de crise na comunidade:

> A gente já tava praticamente desistindo do Noosfero e da comunidade e aí a gente se torna commiter. A gente já tava praticamente decidindo fazer um fork. Só que a gente nem podia decidir fazer um fork, que um fork é uma coisa tão surreal, tão difícil de fazer né um fork definitivo. E era complicado (Entrevistado 5, Commiter, EITA, [63844:65938]).

Refletida por mais de uma organização:

> Chegou um momento que quase a gente faz um fork do negócio, mas como eu sou resistente demais a essa ideia...Internamente no grupo que eu tinha lá no serpro o pessoal sempre quis fazer o fork, eu que fiquei sempre contemporizando, chamava o pessoal lá pra gente discutir, mas hoje eles decidiram coisas que no passado a gente sempre quis, hoje eu sinto uma abertura muito maior pra esse tipo de coisa acontecer porque o pessoal enxergou que se ficasse fechado no mundinho ali da Colivre a coisa não vai ganhar mais desenvolvedores (Entrevistado 15, Commiter, Serpro, [10069:11755]).

A questão da busca por autonomia era importante principalmente pela grande pluralidade de objetivos de uso do Noosfero:

> Uma das motivações pra gente ter mais commiters lá, era algo desse tipo, porque tinha que ter esse trabalho de convencimento grande e como a gente tava caminhando numa direção, tinha outras organizações que caminhavam com outros objetivos, às vezes as coisas não convergiam e isso pra gente atrapalhava bastante. Então esse trabalho de convencimento que tinha que ter foi até reduzido quando a gente teve mais commiters, no caso quando eu virei commiter do noosfero (Entrevistado 13, Commiter, Serpro, [26536:27001]).

Além disso, o custo de centralizar começou a ficar muito alto para a Cooperativa:

> Foi naquele mesmo momento que eles tiveram uma discussão e a comunidade queria bastante que a Colivre flexibilizasse um pouco mais né porque às vezes tinha bastante merge request e tinha poucas pessoas lá na Colivre pra revisar e as coisas acabavam demorando. E aí foi nesse período aí que eles abriram as portas pra mais commiters (Entrevistado 14, Commiter, Colivre, [13120:13600]).

Esses foram portanto os antecedentes para a organização da comunidade e a evolução do seu processo de governança. A Colivre foi muito hábil ao conduzir esse processo pois buscou estabelecer regras claras e transparentes para a entrada de novos commiters e reforçou uma política de desenvolvimento que exigia revisão de código, ou seja, uma segunda opinião para toda proposta nova de código, valendo inclusive para a alta cúpula, os commiters:

> O que a gente fez basicamente foi reunir a comunidade em diversas discussões na lista, em diversas reuniões via videoconferência pra definir um critério de como pessoas se tornam commiters do projeto e de também incluir como commiter pelo menos um dos desenvolvedores principais de cada um desses grupo fora da Colivre. Então, é claro que em um caso que tivesse mais de um desenvolvedor reconhecidamente experiente, seriam dois ou três, quantos fossem necessário. E aí nesse processo a gente definiu essa guia de como uma pessoa se torna desenvolvedor, commiter, o que é que o commiter pode fazer e quem seria o release manager, quais seriam os papéis do release manager, como resolver questões de impasse entre desenvolvedores. A gente definiu mais ou menos uma política né de como as decisões na comunidade são tomadas e aí essas pessoas que eram desenvolvedores de outras organizações passaram a ter poder de commit também no Noosfero. E aí foi quando mais ou menos essa confusão toda se extinguiu aí, quando a gente definiu esse processo (Entrevistado 4, Commiter/RM, Colivre, [48313:51364]).

Inspiradas em experiências de outras comunidades de software livre, essas duas regras de entrada de pessoas e códigos buscavam criar um mecanismo de responsabilização compartilhado. Para a entrada de novos commiters, além do componente mais puramente meritocrático de ter contribuído para o software, dois outros commiters deveriam endossar a entrada do novo membro. Isso criava uma co-responsabilização dos commiters em usar com cuidado esse novo poder. O texto da política dizia: "Eles [os commiters] precisam estar cientes que endossando essas requisições [de novos commiters] significará endossar as ações do novo commiter: se o novo commiter estragar tudo, o endosso dela/dele estragou tudo também" (Política para entrada de novos commiters publicada em 06 de Janeiro de 2015[25]). Para a entrada de código novo, submetido por qualquer commiter, um outro commiter deveria revisar e dar o aceite para que fosse incorporado. Somente se ninguém revisasse dentro de uma semana é que o commiter poderia incorporar seu código diretamente. Como uma organização tinha em média apenas um commiter, a probabilidade dessa revisão ser feita por um commiter de outra organização era grande. E com isso se criou também, nesse caso de forma não intencional, um mecanismo de co-responsabilização entre organizações, diminuindo os riscos de que novas atualizações quebrassem casos de uso de outras organizações:

> Eu aprovei bastante código. Era mais fácil aprovar código de outros commiters né. Então, no começo eu conversava muito com o [nome do commiter]. "[nome do commiter], dá uma olhadinha nesse código aqui, é importante e tal!" O [nome do commiter] é um diálogo muito simples, muito fácil, ele aprovava código meu, eu aprovava código dele. Então, a gente fez um pouco esse intercâmbio no início pra gente ter um pouco mais de credibilidade. Mas depois não foi necessário mais isso. A gente mesmo aprovava o nosso código depois de uma semana. [...] Então, a confiança foi crescendo e o projeto foi crescendo (Entrevistado 5, Commiter, EITA, [44326:45620]).

Essa transição segura, viabilizada pela atualização da governança da comunidade, foi importante para mitigar os temores da Colivre em relação a perda do controle do Noosfero e os riscos de que uma nova correlação de forças inviabilizasse a sustentabilidade financeira da cooperativa.

Na medida que as organizações foram exercendo sua autonomia, elas também foram se afastando umas das outras. Isso gerou uma organização praticamente sem encontros para

[25] Disponível em: https://gitlab.com/noosfero/noosfero/blob/2bdb49acc2f9faa5e88a94b256c9359b2366a165/DEVELOPMENT.md

tomar decisões em conjunto. Como cada organização tinha seu commiter e o prazo para incorporação sem revisão era de apenas uma semana, o Noosfero acabou se transformando numa federação de organizações, cujo único ponto de contato era o uso do mesmo software:

> Então, de certa forma não existia uma centralidade das decisões, sabe! As decisões, elas eram tomadas em lugares separados e aí depois existia um esforço de dicas todas entrassem ali sem causar impacto nas outras funções né. Então, de certa forma a coisa ficou meio independente, sabe, a medida que as lideranças simplesmente acordavam em não atrapalhar umas às outras. Se você não atrapalhava ninguém, você podia fazer o que você quisesse. Era basicamente isso sabe! (Entrevistado 5, Commiter, EITA, [18134:19964]).

De tempos em tempos surgiam iniciativas com o objetivo de melhorar a comunicação entre as organizações, mas eram apenas encontros informativos. Nenhuma decisão era tomada ali:

> Essa centralidade de decisões foi se perdendo. Hoje retomou essas reuniões de comunidade né, entre vários atores, não somente da Colivre como era no início. Mas é só pras pessoas se sincronizarem, sabe, pra elas falarem o que estamos fazendo e tal. Bate alguma coisa: "opa, eu to fazendo isso e você também ta fazendo isso". Então, é só um encontro, não é uma tomada de decisões mesmo (Entrevistado 5, Commiter, EITA, [18134:19964]).

Em resumo, a insistência das lideranças das organizações da comunidade Noosfero em disputar seu espaço de autonomia levou a Colivre a abrir mão de poder - de uma forma bastante hábil - ao mesmo tempo que a autonomia obtida viabilizou a revelação de lideranças fortes em cada organização, cada uma focada nos seus objetivos específicos, mas investindo juntas no aprimoramento do mesmo comum, o Noosfero. Essa foi portanto, a contribuição da comunidade para a façanha do Noosfero.

4.1.3. Fatores tecnopolíticos

No contexto de uma comunidade de desenvolvedores com lideranças eminentemente técnicas, a comunidade buscou solucionar alguns problemas de governança com o uso de tecnologia. Estamos considerando que o fator tecnopolítico está presente quando a comunidade desenvolve recursos tecnológicos para atender ou aprimorar alguma regra de governança. A aplicação de tecnologia para esse fim depende de um contexto institucional que propicie esse tipo de intervenção. O fato das comunidades FLOSS irem desenvolvendo

sua governança ao longo do tempo faz com que a maior parte do tempo elas sejam campo fértil para a atuação criativa de seus membros na interferência para a construção dessas regras, ou seja, na construção ativa da própria instituição. Da mesma forma que o conceito de "entrelaçamento institucional" trazido por Abers & Keck (2013) nos diz que quando a incerteza sobre a vigência das regras e as competências e fronteiras entre os agentes criam um ambiente propício para a criatividade política, podemos dizer que o processo de amadurecimento das comunidades FLOSS e especialmente aquelas que no caso do Noosfero é formada por uma pluralidade de organizações, simulam esse mesmo ambiente.

O primeiro desses fatores foi a construção de uma infraestrutura de plugins no Noosfero. Fazendo um paralelo, é mais ou menos como funcionam atualmente os nossos celulares que, embora sejam fabricados pela Google ou pela Apple, permitem a instalação de um sem número de aplicativos de terceiros sem que essas grandes fabricantes tenham que acompanhar ou participar do desenvolvimento de cada aplicativo. O único requisito é que os aplicativos sigam um conjunto de regras gerais, independente de como seu código está desenvolvido. No Noosfero, a construção desta infraestrutura foi um importante fator tecnopolítico para dar autonomia aos commiters e evitar a divisão da comunidade em grupos menores. A priorização e criação de uma infraestrutura de plugins pela comunidade não trouxe nenhum benefício direto para o uso do software em si, pois tudo que foi desenvolvido via plugin poderia ter sido desenvolvido da maneira tradicional. Por outro lado foi um recurso fundamental para que usos tão diversos do software coexistissem dentro da mesma comunidade utilizando a mesma versão do software:

> Porque ele [o Noosfero] nunca foi desenvolvido como um projeto, como um produto, ele foi desenvolvido pra atender demandas específicas de clientes, de pessoas que queriam usar o Noosfero, então eu diria que 90% do desenvolvimento que foi feito no Noosfero foi assim. É uma empresa ou uma organização que quer atender uma demanda de algum cliente ou de algum sei lá grupo e aquela pessoa fala: "Eu quero usar, eu quero um ambiente onde eu possa publicar coisa num mural". Então, aquela pessoa demanda isso e aí o desenvolvedor que tá envolvido com a comunidade fala: "Dá pra fazer o que esse cara quer no Noosfero, mas não tem essa funcionalidade de mural né, então, eu vou implementar isso". E aí, isso sendo feito pra interesses diversos. Então o Noosfero, ele é usado na área de educação, na área econômica né, na área de participação social, na área de intranet, o diabo a quatro. Então, são diversas coisas diferentes que vão entrando, que tem que dialogar umas com as outras (Entrevistado 4, Commiter/RM, Colivre, [83062:84603]).

O recurso tecnológico da infraestrutura de plugins permitiu simplificar a aceitação de código externo, quando essa decisão ainda era concentrada na Colivre e viabilizou o desenvolvimento das regras de governança de autonomia para as organizações da comunidade. A própria revelação de novas lideranças pode ter sido facilitada - pra não dizer viabilizada - por esse fator tecnológico. Há relatos de que foi essa estrutura que permitiu um processo de individualização das organizações como base para a construção de uma federação com líderes fortalecidos:

> Uma prática que se tornou muito comum no Noosfero e que tinha problema também, era fazer tudo como plugins né, ou seja, cada um no seu quadrado. Basicamente, aquela música do quadrado. Cada um mexe no seu quadrado, no seu plugin. Os quadrados não se juntam né. Os quadrados não se misturam, não tem um quadrado em cima do outro. Então, isso permaneceu por muito tempo né, cada um fazia o código no seu plugin. Por causa disso né, era razoavelmente raro um quebrar o código do outro. Então, a primeira prática era o meu código, eu tô fazendo um código, vou fazer ele como plugin, vou ter certeza que eu não tô mexendo no código de outras pessoas. E quando eu mexer, é uma mudança que eu tenho certeza pra não dar problema no quadrado do outro, no core do Noosfero. Então, por muito tempo ficou isso, até depois de bastante tempo cada um no seu quadrado, cada um no seu indivíduo fortalecido, ou seja, cada um tranquilo, cada um mexendo no seu quadrado, aí teve um amadurecimento da comunidade. Aí sim a comunidade começou a conversar mais, ou seja, é um caminho interessante a gente ver que a comunidade, ela só se constrói depois que os indivíduos estão muito fortes. Se os indivíduos não tiverem fortes, não existe coletivo, sabe. Então, depois que o Serpro podia fazer muita coisa no Noosfero, depois que o Cirandas podia fazer muita coisa no Noosfero, depois que a Colivre e assim por diante (Entrevistado 5, Commiter, EITA, [38991:41021]).

Em resumo, há fortes evidências de que esse foi um fator tecnopolítico central para a permanência de um único software mantido pela Federação Noosfero, em detrimento de um estilhaçamento em 3 ou 4 softwares livres diferentes e sem comunidade, o que provavelmente seria o destino do Noosfero se ele se dividisse numa guerra fratricida. Num cenário sem a infraestrutura de plugins, haveria muito menos meios concretos para conciliar os interesses diversos de cada organização, que ao naturalmente puxar o desenvolvimento para o lado do seu próprio objetivo acabaria por inviabilizar os objetivos das outras.

Um outro fator tecnopolítico importante para o desenvolvimento da comunidade foi criar uma estrutura automática para testagem de código. Como não nos interessa aqui os efeitos disso na qualidade do Noosfero e eventuais diminuição de bugs, vamos focar nossa análise nos efeitos que essa estrutura tecnológica teve na governança da comunidade. Uma

das críticas mais recorrentes dos desenvolvedores ativos periféricos do Noosfero, principalmente na época que a Colivre centralizava a incorporação de código, era a baixa flexibilidade em aceitar código feito por desenvolvedores externos à cooperativa, utilizando critérios que não eram reconhecidos como legítimos por esses desenvolvedores. Veremos mais adiante que muitas dessas justificativas soavam razoáveis para os de dentro da cooperativa mas ao mesmo tempo pareciam demasiadamente exageradas na visão dos desenvolvedores externos. A recusa de código por esses motivos era feita evocando a meritocracia e uma política técnica que não era consensual na comunidade. A questão da meritocracia soava como uma exigência inatingível de perfeição, além dos questionamentos políticos que a meritocracia suscita:

> O [commiter] falava né que o software livre era meritocrático né, o principal critério dele, quase que o critério exclusivo dele era meritocrático e eu combati muitas vezes, eu mandei e-mails até criticando diretamente a meritocracia. "Não dá pra ser apenas meritocrático". É uma ilusão achar que funciona apenas pela meritocracia. Engraçado né, porque meritocracia é um pensamento mais de direita e o pessoal da Colivre tinha um alinhamento mais de esquerda. Apesar de o alinhamento mais de esquerda, na prática, no trabalho, eles acreditavam na meritocracia" (Entrevistado 5, Commiter, EITA, [35374:38150]).

Apesar da ambivalência entre a ideia de meritocracia técnica bastante presente no software livre e as práticas comunitárias de inclusão e intercâmbio de conhecimento também inerentes ao Software Livre - ambivalência essa que não vamos entrar aqui - havia uma intenção legítima das lideranças da Colivre em garantir a qualidade do Comum Noosfero e consequentemente a longevidade da comunidade. Por outro lado não havia nenhum instrumento mais objetivo para ajudar a aferir se aquele código iria prejudicar o Noosfero no longo prazo ou não. Com a inexistência desse instrumento, as argumentações eram construídas a partir da experiência pessoal dos desenvolvedores que tomavam a decisão de recusar o código e, como vimos, suas justificativas não eram aceitas por parte da comunidade.

O desenvolvimento desse instrumento foi, portanto, o fator tecnopolítico que analisamos aqui. Insatisfeito com as constantes recusas de código acompanhadas por discussões infinitas sobre política técnica, um desses desenvolvedores periféricos (à época pois posteriormente ele viraria commiter), contando inicialmente com a ajuda de alguns outros dissidentes, construiu, de forma integrada à plataforma de codificação social um validador de código que informaria publicamente todos os desenvolvedores do Noosfero se

aquela atualização tem qualidade suficiente para ser integrada ao Noosfero sem gerar problemas futuros. Esses validadores eram baseados em iniciativas populares à época que já eram utilizadas por algumas das grandes comunidades FLOSS:

> Eu investi muito tempo pra colocar três sistemas de testes automatizados, todos eles rodando ao mesmo tempo, todos eles rodavam a mesma switch de testes né. É o Trevis CI que é muito comum no mundo open source, o maior sistema de testes automatizados open source. O gitlab CI, que é integrado ao gitlab e hoje é o padrão do Noosfero e o Circle CI. Aí quando eu consegui manter isso e aí toda hora que o teste quebrava, eu reclamava do pessoal do Noosfero: "Oh quebrou o teste, quem quebrou foi tal pessoa." Aí eu comecei a policiar um pouquinho isso daí porque toda vez que entrava um código novo e ele não falhava os testes, a chance de ele entrar pro código era praticamente 100%. E aí tinha uma maneira fácil né, simplesmente a pessoa mandava o código e automaticamente todos os testes eram rodados e dava pra você provar: "oh eu não quebrei nada, ta tudo funcionando ainda". Então, era muito fácil ver que a pessoa não fez nenhuma merda, sabe! (Entrevistado 5, Commiter, EITA, [11889:14468]).

E o instrumento vai evoluindo e continua fazendo sentido mesmo depois da reorganização da comunidade com a promoção de grande parte dos desenvolvedores ativos periféricos para commiters:

> Com o tempo, à medida que eu fui melhorando a estrutura de testes, essa estrutura de testes eu já fui mexendo automaticamente, porque eu não mexia no Noosfero em si, mas simplesmente uma ferramenta pra testar o Noosfero. À medida que eu fui melhorando ela, eu pude criar a confiança própria de me manter como commiter e ser um bom commiter, digamos assim né. Um cara que não faz merda no código né. Tinha aquela insegurança muito grande né, "Poxa, eu tô como commiter agora, mas eu sei que os caras são meio chatos, então… É como se eu fosse um commiter, mas eu ainda não sou porque eu não tenho confiança ainda. Então, eu tive de desenvolver um método de provar que o meu código era de confiança e esse método foi a questão dos testes, sabe. Então, eu mesmo tive de desenvolver um mecanismo de ter confiança dos outros, sabe, e não existia esse mecanismo (Entrevistado 5, Commiter, EITA, [29774:31180]).

O mecanismo foi aceito pela comunidade e posteriormente integrado na política de desenvolvimento (Entrevistado 15, Commiter, Serpro, [22015:23694]), tendo contado inclusive com a participação da Colivre no seu desenvolvimento (Entrevistado 4, Commiter/RM, Colivre, [84622:85231]). Com a adoção dessa infraestrutura tecnopolítica pela comunidade, todo código que não passasse nos testes seria automaticamente recusado. E código que passasse, não seria mais recusado, pelo menos não por motivos meritocráticos intrínsecos ao código. Ele ainda poderia ser recusado se prejudicasse o uso de outras

organizações, mas esse não era o principal dilema para a crise de recusa de código pela qual a comunidade passava.

Praticamente em paralelo com a entrada do fator tecnopolítico da infraestrutura de testes houve duas mudanças na comunidade Noosfero. A primeira foi a criação das regras de promoção de commiters e a consequente ampliação do núcleo de desenvolvedores centrais. A outra mudança foi o distanciamento de uma das lideranças da comunidade, a qual foi atribuída boa parte da inflexibilidade na aceitação das contribuições externas. Por conta dessas mudanças, nunca saberemos se esse fator tecnopolítico teria conseguido equilibrar o poder de incorporação, favorecendo os desenvolvedores periféricos. Por outro lado há evidências de que essa infraestrutura contribuiu para o fortalecimento da confiança e autonomia dos desenvolvedores líderes das organizações. Dessa forma, junto com a infraestrutura de plugins, o fator tecnopolítico foi muito importante para preservação da união da Federação Noosfero. Foi portanto, em boa medida, também responsável pela façanha do Noosfero.

Trabalhamos aqui portanto, a hipótese de que a criatividade dos atores na construção de soluções tecnopolíticas que atendessem suas necessidades na relação com o conjunto da comunidade foi um fator relevante para a manutenção da unidade. Adicionalmente, nos parece que esse tipo de agência criativa também deve o seu sucesso ao ambiente entrelaçado que a comunidade FLOSS do Noosfero configura, pois do contrário, num contexto onde as regras de governança fossem mais definidas e com pouco histórico de mudanças, esse tipo de iniciativa tecnopolítica encontraria resistência a ser adotada pela comunidade. Num contexto mais rígido, talvez nem houvesse muitos incentivos para a própria criatividade dos agentes, eventualmente sendo mais fácil aplicar essa criatividade num racha com a comunidade, colocar o código embaixo do braço e começar uma nova instituição.

Na tabela 2 abaixo, há um resumo analítico dos três conjuntos de fatores analisados aqui: organização central (Colivre), protagonismo da comunidade e os dois fatores tecnopolíticos. Nesse resumo, evidenciamos como cada um desses fatores influenciou na tomada de decisão e contribuiu para a façanha da comunidade Noosfero.

Fatores que influenciaram na tomada de decisão e contribuíram para a façanha da comunidade Noosfero ter permanecido unida		
Nome do fator	Como influenciou na tomada de decisão	Como contribuiu para a façanha
Fator “uma organização central”	Concentrou poder: Centralizou as decisões sobre o código exigindo que os desenvolvedores periféricos se adequassem à política técnica da Cooperativa	Nutriu a comunidade: Trabalhou ativamente para que o Noosfero se tornasse um comum, com publicação regular de versões e disponibilização do software para a comunidade
Fator “protagonismo da comunidade”	Disputou poder: Produziu uma crise na comunidade exigindo que a organização central abrisse mão de poder e criasse regras claras para os desenvolvedores periféricos se tornarem desenvolvedores centrais	Revelou lideranças: Com o poder distribuído, as organizações puderam formar e revelar lideranças que dinamizaram a comunidade e contribuiram para o aprimoramento do Noosfero como um comum
Fatores “tecnopolíticos”	Descentralizou poder e gerou confiança: As soluções tecnopolíticas permitiram que os desenvolvedores periféricos atendessem suas necessidades através de uma alternativa fora da governança formal. O contexto de “entrelaçamento institucional” das comunidades FLOSS viabilizou a aplicação desse tipo de criatividade pelos atores	Manteve a unidade: A criatividade política dos agentes foi aplicada na construção de soluções tecnopolíticas, que ao serem adotadas e colocadas à disposição dos membros, atendeu as necessidades das organizações e os conflitos não chegaram a gerar a divisão da comunidade em vários softwares distintos

Tabela 2: Resumo analítico dos três conjuntos de fatores que influenciaram na façanha da comunidade Noosfero
Fonte: Produção própria

4.2. Os personagens e seus poderes

Nesta seção, analisaremos os comportamentos identificados pela pesquisa e como cada comportamento contribuiu para o aumento de influência do membro da comunidade nas decisões sobre o software. Falaremos portanto, dos personagens da comunidade e suas

diferentes formas de obter mais autoridade progredindo em direção às áreas mais centrais da comunidade.

Como já expusemos (ABERS & KECK, 2013), as práticas de construção institucional que ajudam a acumular autoridade prática passam pelo engajamento com outros atores (aumento da conectividade) e pela realização de experimentação, recombinando ideias, recursos e relacionamentos. Essas práticas produzem capacidades e reconhecimento, fundamentais para o acúmulo de um certo nível de autoridade prática. Complementarmente, a progressão de autoridade lateral descrita por DAHLANDER & O'MAHONY (2011) acontece quando os membros do projeto se movem em direção ao centro, tendo contato com mais áreas do projeto. Esse fenômeno é distinto àquele conhecido como subir na carreira ou ser promovido na firma, pois ao invés de passar a gerenciar mais pessoas, no caso das organizações planas os indivíduos passam a ter responsabilidade e direitos de decisão sobre uma porção maior do trabalho coletivo sem que isso implique em supervisionar mais pessoas. A progressão em direção ao centro seria portanto baseada em aumento de expertise e respeito dos pares e, segundo os autores, seria obtida através de pelo menos três comportamentos:

- Contribuições técnicas: Como as comunidades FLOSS dependem das contribuições técnicas para existir, esse tipo de comportamento aumenta as chances de progressão dos indivíduos para posições de maior poder e responsabilidade;
- Comunicações técnicas: O aprendizado de novos membros é um fator importante para as comunidades, de forma que aqueles membros que possuem mais didática e conseguem responder perguntas e se comunicar enquanto o trabalho está sendo desenvolvido tendem a ganhar mais respeito dos pares e progredir para posições de maior responsabilidade;
- Trabalho de coordenação: Coordenar tarefas interdependentes para que a organização atinja um objetivo comum é um grande desafio para qualquer organização, especialmente aquelas, como as comunidades FLOSS, onde seus membros estão espalhados e fazem parte de diferentes organizações. Dessa forma a realização desse trabalho ganha importância e respeito aumentando as chances de progressão em direção ao centro de quem o realiza.

A leitura da transcrição das entrevistas a partir da técnica de “descriptive coding” foi feita de forma a identificar as três categorias de comportamentos para progressão de autoridade lateral. Posteriormente buscamos identificar se nesses comportamentos também encontramos

as práticas de construção institucional apontadas pela teoria de autoridade prática da ciência política. Além de termos encontrado descrições interessantes bastante sinérgicas com os comportamentos propostos pelos autores, encontramos outras para as quais houve certa dificuldade em encaixar nesses três. Descreveremos esses achados a seguir, incluindo essas novas hipóteses.

4.2.1. Contribuições técnicas

Na comunidade Noosfero, o commiter é uma figura técnica. Isso quer dizer que o requisito principal para alguém se tornar commiter é o de desenvolver código, em detrimento de papeis gerenciais ou de liderança:

> como requisito, requisito mesmo, se tiver uma pessoa que é contribuidor e pede pra ser commiter do noosfero, e é uma pessoa que não quer ter contato gerencial nem líder, mas é uma pessoa que tem capacidade técnica de fazer contribuições, fazer bons códigos e de não quebrar o projeto, ela pode ser commiter tranquilamente. Ela não é obrigada a assumir esses outros papeis (Entrevistado 1, Commiter/RM, Colivre [18505:18974])

Com isso, a trajetória mais comum de todos que se tornaram commiters da comunidade Noosfero é galgada através do comportamento de fazer contribuições técnicas, como nesse exemplo trazido por um dos commiters:

> então eu comecei a participar do projeto com pequenas atividades já dentro do participa que usa o noosfero por baixo, então eu comecei a desenvolver, criar as atividades, fui começando a interagir com a comunidade [...] que mantém o noosfero, fui dando contribuições e fazendo pull requests mesmo no github interagindo no chat e a partir de determinado momento eu terminei virando um dos commiters do projeto, até pelo reconhecimento da comunidade pelas contribuições que eu vinha dando, então eu passei a ter um papel mais ativo e de decisão dentro das funcionalidades do noosfero como um todo e a gente começou a andar de forma mais ativa no projeto. Esse processo demorou vários meses, acho que mais de ano (Entrevistado 13, Commiter, Serpro; [2658: 3587])

A progressão ao centro através do comportamento de fazer contribuições técnicas aparece também no próprio conceito de liderança. Ao falar de um dos desenvolvedores líderes, o Entrevistado 4 conta que

> todo mundo que tava envolvido no projeto, mesmo desenvolvedores e até colaboradores conseguiram reconhecer nele uma figura de líder só pela

> experiência que ele tinha né, até pelo trabalho técnico né que ele fazia né. Então, muitos de nós aprendemos a desenvolver com o trabalho que [ele] executava, então, a gente via o código dele, via como era bem escrito, via como ele tinha realmente uma percepção do projeto diferenciada. Então, querendo ou não, isso acaba trazendo um pouco desse papel de liderança, mas nada que ele chegasse em algum momento pra poder dizer assim: "Não, isso vai ser assim porque eu quero". Então, isso nunca aconteceu (Entrevistado 4, Commiter/RM, Colivre, [70224:72072]).

Um ponto interessante desse depoimento, é que o Entrevistado especifica bem o tipo de autoridade que o líder em questão exercia: uma autoridade que emana da experiência e conhecimento sobre o código. Questionar esse poder implicaria assumir a responsabilidade pela decisão alternativa (ter capacidades para isso) e ao mesmo tempo dependia de angariar apoios através do reconhecimento pelos outros membros. Tanto a capacidade de evoluir o código do Noosfero quanto o reconhecimento de uma certa competência de fazer isso com maestria parecem ser componentes fundamentais da construção da legitimidade dessa liderança. Se por um lado sua autoridade não foi exercida diretamente sobre pessoas, já que suas decisões se concentravam na evolução do Comum de código do Noosfero, devemos matizar isso pelo efeito limitador que esse tipo de decisão produz na autoridade dos outros membros. Em outras palavras, sempre que a liderança decidia sobre uma incorporação de código, ela também limitava as opções de incorporação disponíveis para os outros desenvolvedores, já que não tinham autoridade para reverter o que foi feito. Dessa forma, a autoridade acumulada através do comportamento de contribuições técnicas acabava tendo o efeito de influenciar a atuação dos outros membros a respeito da incorporação de código no Comum mantido pela comunidade. Assim, há indícios de que o comportamento de contribuições técnicas além de levar o membro da comunidade a ocupar posições mais centrais, também parece produzir capacidades e reconhecimento necessários para o acúmulo de autoridade prática. Uma das hipóteses é que a progressão ao centro e o acúmulo de autoridade são duas formas de analisar e comunicar o mesmo fenômeno.

Diante disso, é importante questionar a regra formal que atribui a todos os commiters, sem distinção - já que o RM tem o mesmo poder de um commiter, a mais alta posição de autoridade na comunidade. A aceitação acrítica dessa regra como realidade poderia levar à falsa interpretação de que todos têm o mesmo grau de influência nas decisões sobre o que vai ser incorporado. Ainda que a possibilidade técnica de gravar código na árvore principal

tivesse uma característica horizontal, a diferença aparece na capacidade de influência que alguns desenvolvedores tinham sobre os outros:

> A comunidade sempre foi assim meio que horizontal, em termos de decisão, em relação aos desenvolvedores. Então, todo mundo que é desenvolvedor tá no mesmo nível pra tomar decisão. E o cara que tá entrando agora, que tem um ano de desenvolvedor, como commiter, ele tem o mesmo poder de decisão que [o commiter mais experiente] com 10, 12 anos de missão, né. Claro que os outros desenvolvedores desenvolviam com muito mais atenção, dando muito mais valor a opinião [do commiter mais experiente] que a de um desenvolvedor novo (Entrevistado 4, Commiter/RM, Colivre, [70224:72072]).

Existe portanto um sistema social de contrapeso que limita o poder do indivíduo commiter em prol dos desenvolvedores mais experientes, amparados pelo coletivo. Além da regra de revisão que limita socialmente a possibilidade técnica que um commiter tem de gravar código direto na raiz, e abre a possibilidade de outros commiters questionarem a contribuição antes dela ser aceita, há também limitações mais sutis que vão sendo incorporadas pelos atores no processo de socialização da comunidade. Esse processo de socialização é sentido pelos commiters recém chegados, como nesse exemplo:

> É como se eu fosse um commiter, mas eu ainda não sou porque eu não tenho confiança ainda. Então, eu tive de desenvolver um método de provar que o meu código era de confiança e esse método foi a questão dos testes, sabe. Então, eu mesmo tive de desenvolver um mecanismo de ter confiança dos outros, sabe, e não existia esse mecanismo (Entrevistado 5, Commiter, EITA, [29774:31180]).

Também aqui:

> eu desenvolvia coisas novas, acrescentava, resolvia bugs, mas coisas muito técnicas que demandavam muita experiência e muito poder de decisão, falar vai ser assim por causa disso, era delegado para os commiters mais experientes. A gente meio que seguia o que a comunidade falava ali (Entrevistado 7, Commiter, UnB, [8619: 9557]).

O dia a dia de incorporação de código tende a ser mais fluido pois grande parte das mudanças não afetam o uso que as pessoas e organizações fazem do software. Porém, todo commiter sabe o que deve ser feito quando sua contribuição tem potencial de alterar uma porção mais ampla do software:

> Então, são raros os momentos onde eu como commiter e até os outros como commiter, eu posso dizer, precisam discutir com outros desenvolvedores se aquilo vai entrar ou não. Mas acontece né, em casos onde é uma coisa mais séria, uma coisa que envolve uma mudança mais drástica aí, geralmente essa

> discussão acaba acontecendo na lista (Entrevistado 4, Commiter/RM, Colivre, [30065:31506]).

Quem propõe a mudança deve ter a sensibilidade de saber quando a mudança é merecedora de um trabalho de discussão e convencimento de outros commiters. Essa decisão é influenciada tanto pelo aumento do conhecimento em relação ao código (podendo prever com mais segurança os eventuais impactos indesejados) tanto pela necessidade de obter reconhecimento por parte do coletivo de líderes. Quanto mais alta a reputação nesse coletivo, maior o custo social em quebrá-la. Quanto mais baixa, maior o risco de nunca ser aceito como um líder de fato. Essa condição aumenta consideravelmente o custo social de fazer algo que quebre o trabalho de alguém. Um desenvolvedor se empodera a partir do reconhecimento que recebe desse coletivo de líderes pelo histórico de suas contribuições. E ser reconhecido como um líder nesse sentido, é poder exercer a autonomia de tomar decisões sobre o software naquilo que não afeta o conjunto dos líderes e ter a opinião vencedora nos momentos em que seu trabalho implica em mudanças mais drásticas na base de código da comunidade.

Com isso, podemos concluir que o comportamento de contribuições técnicas é responsável pela progressão de membros da comunidade em direção ao centro. A regra formal que estabelece um desenvolvedor como commiter, ainda que seja um limitador para essa progressão, não é suficiente. Em outras palavras, quando um desenvolvedor se torna commiter ele não atinge o grau mais alto de influência na comunidade, pois isso depende do acúmulo de outros fatores, de capacidades técnicas e algum reconhecimento. Ao contribuir para o acúmulo desses fatores, o comportamento de contribuições técnicas também contribui para o acúmulo de autoridade prática dos líderes, que é exercida por meio de decisões sobre o software, influenciando indiretamente nas opções disponíveis para os membros com menos autoridade ou centralidade na comunidade.

4.2.2. Comunicação técnica

O comportamento de comunicação técnica na comunidade noosfero não era organizado de forma específica, ou seja, não havia membros destacados para essa tarefa. A expectativa é que os membros mais experientes cumprissem esse papel, mas mesmo entre eles havia muita oscilação. Não foi possível coletar evidências de que esse comportamento

influenciava de forma decisiva a trajetória dos membros em direção ao centro. Ao contrário, foi possível perceber alguns indícios de que esse não era um fator importante nesse processo.

> Esse sempre foi um gargalo eu acredito no noosfero, como o volume de trabalho de desenvolver o noosfero de cuidar do noosfero já era grande, não sobrava muito tempo para tirar dúvidas na lista nos canais de comunicação. Essa parte sempre ficou negligenciada (Entrevistado 1, Commiter/RM, Colivre, [38106:39442]).

Como as decisões de incorporação não ficavam a cargo do RM mas sim do grupo de commiters a partir de sua própria dinâmica, seu trabalho era basicamente o de comunicação técnica, concentrado nos momentos das releases. E esse trabalho sempre foi visto pela comunidade como um tipo de fardo, que todos eram obrigado a carregar e por isso deveria haver circulação: "O RM não tem diferença de papel durante o desenvolvimento. ele se iguala ao resto dos core commiters. O RM tem apenas um papel diferenciado que no dia do release ele é o responsável por fazer aquilo tudo" (Entrevistado 1, Commiter/RM, Colivre, [34790:34992]). Além disso,

> era um papel que a gente estimulava que fosse cíclico mesmo, porque muitas das atividades do release manager eram trabalhos um pouco chatos, por exemplo fazer documentação, escrever um release note com as mudanças e as correções que foram feitas naquela versão, então tinha algumas atividades que eram meio enfadonhas, então a gente acabava tendo uma prática de esse bastão passar de uma pessoa pra outra (Entrevistado 1, Commiter/RM, Colivre, [4036:4439]).

O argumento aqui não é de que esse tipo de contribuição não tinha efeito nenhum. Desenvolvedores que se dedicassem a esse tipo de trabalho ganhavam sim reconhecimento da comunidade. O que não foi possível perceber de forma explícita é de que maneira isso implicava em ter mais influência nas discussões e decisões sobre incorporação de código.

Por outro lado, a principal liderança da comunidade era vista não somente como alguém que sabe muito ("é o cara que faz funcionar, que resolve, é aquela coisa do software livre que fora do software livre não é muito bem vista, a meritocracia, é a questão da meritocracia", Entrevistado 1, Commiter/RM, Colivre, [33238:33780]) mas também como alguém que tinha didática (Entrevistado 2, Commiter/RM, Colivre, [3815:4129]), sabia como passar o conhecimento: "e ele além de saber muito ele sabia passar fácil, ele conseguia passar fácil, não era só questão de saber era saber e compartilhar isso com o grupo" (Entrevistado 1, Commiter/RM, Colivre, [47070:47217]). A força dessa liderança ficou bastante evidente tanto

na fala dos que o acompanhavam (“faz muito sentido pros meus ouvidos as coisas que ele falava os argumentos que ele usa, e não é muito convincente no sentido de estratégia de discurso é convincente mesmo de conteúdo”, Entrevistado 1, Commiter/RM, Colivre, [33238:33780]), quanto na fala dos que o confrontavam:

> na verdade é assim, a galera enxerga como um pessoa técnica muito competente, e isso é inegável, mas ele não é o cara que acerta tudo em todos os contextos. Endeusava[-se] muito [ele], tudo que ele falava virava palavra de Deus (Entrevistado 15, Commiter, Serpro, [37423:38510]).

A hipótese é, portanto, que o comportamento de comunicação técnica não foi suficientemente relevante como mecanismo de progressão ao centro da comunidade se visto isoladamente. Porém como um componente adicional ao comportamento de contribuições técnicas teve um papel preponderante em sedimentar uma importante liderança, fornecendo justificativas plausíveis para aqueles que defendiam seus ensinamentos: além de resolver o problema (meritocracia) ele justificava muito bem. Do ponto de vista do acúmulo de autoridade prática, parece mais plausível que o comportamento de comunicação técnica pode ter atuado como um importante vetor de reconhecimento para quem já tinha capacidades técnicas, ou seja, para os contribuidores mais ativos. Se olharmos dessa maneira, a habilidade de comunicação técnica parece ter sido menos uma capacidade independente e valorizada pela comunidade e mais um fator de reconhecimento para quem já tinha acumulado capacidades mais relevantes.

4.2.3. Trabalho de coordenação

O modelo de governança da comunidade Noosfero - e talvez essa seja uma hipótese passível de generalização para grande parte das comunidades de software livre - foi sendo desenvolvido conforme a necessidade da própria comunidade. Recapitulando a história, a comunidade Noosfero nasce pelo ato criador de uma cooperativa de software da Bahia, a Colivre, com o objetivo de criar um software livre que atendesse a demanda de dois clientes distintos, uma rede internacional focada em temas de internet e uma rede nacional de economia solidária, o FBES (Fórum Brasileiro de Economia Solidária). O primeiro cliente acaba não vingando mas a rede de economia solidária passa a ver no Noosfero uma tecnologia estratégica para os coletivos de economia solidária e em determinado momento viabiliza a

entrada de uma cooperativa responsável por desenvolver e manter as tecnologias utilizadas pelos coletivos, a cooperativa EITA (Cooperativa de Trabalho Educação, Informação e Tecnologia para Autogestão). A EITA é, portanto, uma das primeiras organizações, além da Colivre, a entrar com força de desenvolvimento própria para Noosfero. É o primeiro momento de ampliação da comunidade. Com o passar do tempo, o Noosfero começa a ser adotado por outras organizações que também se organizam para desenvolver código, como a Universidade de São Paulo (USP), através do Centro de Competência em Software Livre (CCSL), a Universidade de Brasília (UnB), através do curso de Engenharia de Software da Faculdade do Gama (FGA) e finalmente a maior empresa pública de tecnologia do Brasil, o Serviço Federal de Processamento de Dados (Serpro). A ampliação de organizações aptas a desenvolver código para o Noosfero não é acompanhada por uma atualização nas regras de governança, que segue durante um tempo centralizada na cooperativa fundadora, a Colivre. Esse controle do Noosfero pela Colivre se dava de duas formas, a primeira por concentrar a nomeação do RM e a segunda por ter em seus quadros, durante quase todo o período de vida da comunidade, o desenvolvedor mais experiente e influente do software.

Como já vimos, o RM concentra o trabalho de coordenação necessário para garantir a vida do software, através da publicação das releases. Esse é um trabalho vital para a comunidade, pois um software sem release é como um jornal que nunca é impresso/publicado. É um software morto, pois não se atualiza e nem desperta interesse de organizações e desenvolvedores, seja para utilizar, seja para contribuir com ele. Esse trabalho envolve definir quais serão as atualizações que irão compor uma determinada release, porém já vimos que essa não era uma decisão do RM, mas sim do grupo de commiters, ao longo do intervalo de tempo entre uma e outra release. Muitas vezes uma nova atualização traz consigo a necessidade de um novo componente de software, como já vimos, as chamadas bibliotecas. Segundo o desenvolvedor mais influente do grupo, o RM teria a palavra final em relação a adoção desses novos componentes, ainda que essas discussões envolvessem todo o grupo de commiters, segundo seu grau de influência:

> [A preocupação do RM] é garantir que as features, que as funcionalidades aceitas no código, elas fazem sentido, são viáveis de ser mantidas no futuro né. Às vezes, tem coisas que parecem uma ótima ideia, mas na prática não vai ser viável manter aquilo no futuro. Então, tem que garantir a manutenibilidade do projeto (Entrevistado 11, Commiter/RM, Colivre, [32726:33191])

Dessa forma, ainda que o RM não tivesse a prática de impor sozinho uma decisão, já que isso dependia de um certo suporte por parte do grupo de commiters, o fato da Colivre sempre ter nomeado o RM do Noosfero pode ser caracterizado como uma espécie de bastão diplomático, fazendo com que todos os commiters e desenvolvedores das outras organizações que compunham a comunidade tivessem que negociar com quem de fato tinha a palavra final. Mesmo assim, não há indícios de que os desenvolvedores da comunidade, principalmente os de fora da Colivre, tenham em algum momento disputado essa posição. A disputa se deu em torno de ocupar posições de commiter, que já seriam suficientes para que a organização tivesse código incorporado, independente das prerrogativas do RM, que como vimos, na prática eram de ordem operacional. Inclusive, o atual RM da comunidade acredita que o motivo dele exercer esse papel é simplesmente por ser o desenvolvedor mais ativo, e não por um eventual reconhecimento quanto ao trabalho de coordenação (comportamento que aqui analisamos): "E eu acho que eu continuo sendo release manager justamente por causa disso, talvez seja mais uma relação inversa né, seja uma relação de eu fazer isso porque eu sou ativo" (Entrevistado 4, Commiter/RM, Colivre, [63016:64492]).

A primeira evolução de governança da história do Noosfero acontece com a abertura do grupo de commiters para fora da organização principal, a Colivre. Ainda que esse seja um caminho esperado para qualquer comunidade, ele não aconteceu de forma automática, tendo exigido uma certa mobilização das organizações e desenvolvedores externos para que isso acontecesse. O primeiro passo foi o reconhecimento do gargalo:

> o gargalo era esse, não tinha uma regra clara, e era muito difícil se tornar commiter, existia uma certa resistência, não sabia o que ia acontecer, e durante muito tempo só o pessoal da colivre que eram commiters, e aí a gente começou a ter gente do Serpro, gente da UnB tendo contribuições importantes, e aí a colivre teve que realmente abrir pra comunidade esse processo (Entrevistado 3, Desenvolvedor comum, USP e UnB, [25977:28070]).

Ainda que o gargalo fosse reconhecido pela Colivre, o principal dilema era o fato do Noosfero sempre ter sido um FLOSS comercial, ou seja, além de um comum disponível para uma comunidade ativa de organizações, era também a principal fonte de renda da Colivre. Os limites dessa abertura esbarravam portanto nas incertezas sobre o futuro do software e dos custos de manter o projeto, caso a Colivre eventualmente se tornasse minoritária no novo arranjo de governança:

> No momento em que outros agentes começaram a se envolver na Comunidade, existia um certo receio da Colivre enquanto organização mesmo de simplesmente abrir o projeto para outras pessoas poderem contribuir e commitar coisas e modificações, com receio por conta de esse processo de abertura acabar prejudicando a capacidade da Colivre de se sustentar pelo projeto, que é um receio compreensível né, pelo menos do meu ponto de vista. Até você ter um plano concreto de como isso pode acontecer. Então, esse tempo que a Colivre levou pra construir, junto com os agentes da comunidade, um modelo onde todos pudessem participar ativamente de forma igualitária do projeto, sem que a Colivre sentisse aquele receio de que a qualquer momento um daqueles agentes poderiam acabar destruindo o projeto e ela se prejudicar e, digamos assim, morrer como empresa por não ter mais como se sustentar com ele, levou tempo né. E esse tempo para muitas pessoas na época foi demais né (Entrevistado 4, Commiter/RM, Colivre, [48313:51364]).

Esses dados levantam a hipótese de que o trabalho de coordenação não parece ser um fator relevante para a progressão ao centro da comunidade Noosfero. Essa hipótese é embasada pelo fato de que o primeiro salto para evolução da governança da comunidade acontece como fruto de uma pressão organizada pelos desenvolvedores das outras organizações que faziam parte da comunidade e a busca não era por ocupar a posição principal onde o trabalho de coordenação se concentrava (o RM), mas sim pela autonomia em incorporar código no ramo principal e assim diminuir os riscos da organização se distanciar da base de código comum. Após esse primeiro salto na governança do Noosfero, um dos commiters externos desenha para nós qual era o novo status-quo que se estabeleceu com esse movimento:

> é obrigação de quem tá propondo o código articular e mobilizar os commiters né. Então como a gente tinha bastante commiters, - era um commiter em cada organização - as pessoas que entravam com algum código, normalmente elas tinham alguém próximo delas, para incorporar o código delas, um colega de trabalho por exemplo. [...] Então, a gente tinha meio que um commiter em cada canto, sabe! Então, a gente viu que não tinha mais esse problema de centralizar o trabalho de integração de código. Não. Cada organização se organiza e integra o código porque tem uma pessoa de confiança, tem um commiter ali (Entrevistado 5, Commiter, EITA, [43005:44287]).

Complementarmente, o trabalho de coordenação não parece ser uma capacidade relevante para os membros obterem acúmulo de autoridade prática. Ao mesmo tempo não parece contribuir para o aumento do reconhecimento dos membros, pelo menos não no nível necessário para produzir acúmulo suficiente de autoridade prática a ponto de produzir

influência nas ações dos outros desenvolvedores. Esse achado pode estar relacionado ao fato de, como já vimos, a comunidade Noosfero se parecer com uma federação de organizações em torno de uma base comum de código, em vez de uma comunidade com forte organicidade própria. A dificuldade em encontrar evidências para o trabalho de coordenação como um vetor de incremento ou acúmulo de autoridade lateral e prática dos membros pode estar ligada a essa característica específica. Quando o vínculo organizacional mais forte não é a comunidade, esse tipo de trabalho confere menos reputação a quem o realiza, produzindo menos incentivos.

Na tabela 3 abaixo apresentamos uma sistematização dos achados em relação aos três comportamentos analisados, os de "Contribuições técnicas", "Comunicação técnica" e "Trabalho de coordenação" e como eles influenciam a progressão ao centro da comunidade bem como o acúmulo de autoridade prática.

Suficientemente teorizados (DAHLANDER & O'MAHONY, 2011)			
Comportamento	Breve descrição	Força na comunidade Noosfero (Autoridade Prática de ABERS & KECK, 2013)	Limites percebidos
Contribuições técnicas	Faz trabalhos de produção de código e toma (ou participa na tomada de) decisões arquiteturais sobre o software	Muito forte: principal comportamento para a progressão ao centro e fazer parte do grupo de commiters. Contribui para a obtenção de capacidades técnicas e algum reconhecimento pelos outros commiters (acumula autoridade prática)	Se exercido isoladamente não garante poder de influência entre os commiters já que requer o comportamento de "comunicação técnica" para obter reconhecimento da comunidade
Comunicação técnica	Faz trabalho de formação de outros membros e explicações didáticas sobre os aspectos técnicos do software	Fraca: É relevante na comunidade mas não funciona isoladamente. Pode ter funcionado mais como vetor de aumento de reconhecimento do que como uma capacidade relevante para acúmulo de autoridade prática	É fator essencial para obter níveis maiores de influência entre os commiters mas isoladamente não garante acesso ao espaço central da comunidade nem ao acúmulo de autoridade prática

Trabalho de coordenação	Coordena as atividades e contribuições de todos para que a comunidade avance e atinja um objetivo comum	Fraca: Visto como um fardo, aumenta o respeito mas só funciona se associado com o comportamento de contribuições técnicas. Por não ser visto como uma capacidade relevante, não produz acúmulo suficiente nem reconhecimento, gerando um efeito mais rarefeito no acúmulo de autoridade prática	Como só pode ser exercido por membros que já fazem parte da área central, isoladamente não contribui para a progressão ao centro

Tabela 3: Como os três comportamentos de autoridade lateral influenciam na progressão ao centro da comunidade Noosfero bem como o acúmulo de autoridade prática

Fonte: Elaboração própria

4.2.4. Hipóteses complementares

Além da análise dos três comportamentos que, segundo a teoria, contribuem para a progressão dos membros em direção ao centro da comunidade (contribuições técnicas, comunicação técnica e trabalho de coordenação), buscamos identificar outras possibilidades que não foram suficientemente cobertas por esses três comportamentos já teorizados. O objetivo aqui não é concluir pela existência de comportamentos com o mesmo nível de generalização, mas lançar hipóteses extras que ajudem em pesquisas futuras sobre o tema da autoridade lateral em comunidades de software livre, especialmente lidando com comunidades menores e com características similares com a comunidade do Noosfero aqui estudada, especialmente aquelas que lutam diariamente pela sobrevivência.

No mundo do software livre, algumas empresas utilizam uma estratégia conhecida de contratar os desenvolvedores mais ativos ou até mesmo os commiters (mantenedores) de um projeto para influenciar as decisões sobre o software: “na prática, quem participa da vaquinha é quem decide o que é que vai ser feito, né!” (Entrevistado 11, Commiter/RM, Colivre, [11099:12525]). Na história do Noosfero isso não aconteceu diretamente, porém durante toda a vida da comunidade, diversas organizações aderiram ao software através de projetos que

viabilizaram o financiamento do desenvolvimento. As pessoas que atuavam nessas organizações não passavam a fazer parte automaticamente da comunidade. Apenas aquelas que faziam parte de equipes próprias de desenvolvimento, como foi o caso da Cooperativa EITA, a Faculdade do Gama (FGA/UnB) e o Serpro. Em alguns desses casos, os principais commiters do Noosfero eram convidados a compor a equipe dos projetos, com o objetivo de orientar os demais desenvolvedores e simplificar a comunicação para incorporação de código que era de interesse dos projetos. Esse artifício foi utilizado tanto pela Cooperativa EITA quanto mais fortemente pelo projeto do CCSL/USP e da FGA/UnB. A cooperativa que fundou o Noosfero, a Colivre, ela mesma uma organização que sobrevivia financeiramente do Noosfero, reconhece a importância da relação com esse tipo de ator, pois é o que garante a sustentabilidade do software: "assumir que Software Livre é feito puro e simplesmente pelo bem da humanidade e que todo mundo quer cantar em volta de uma fogueira e ser feliz… Então é um pouco ingênuo" (Entrevistado 11, Commiter/RM, Colivre, [11099:12525]).

A partir desse contexto foi possível perceber a atuação de alguns membros da comunidade Noosfero que se tornaram referência para a comunidade pela capacidade de captar e gerir projetos envolvendo o software. Esse comportamento levou ao reconhecimento de pelo menos um desses membros como uma liderança da comunidade, ao mesmo tempo que suas contribuições técnicas foram mínimas. Aqui fica bem evidente como essa atuação se dava no contexto da comunidade e não como uma organização externa:

> Então geralmente, pra deixar bem claro, geralmente botava o interesse da comunidade e daquilo que era necessário pro noosfero independente do interesse das coisas que eu tava a frente. Então eu sempre tive esse comportamento de, eu era mais, eu não separava, eu era um cara da comunidade que tava com aquele recurso na mão, eu nunca deixei de me apresentar como alguém da comunidade noosfero, mesmo eu coordenando projetos grandes eu era um cara da comunidade que tava com aquele recurso na mão e o que era possível eu alocava pra evolução do noosfero (Entrevistado 3, Desenvolvedor comum/Articulador de Recursos, UnB, [21360:22515]).

Além de aumentar as chances de ser reconhecido como uma liderança da comunidade, esse tipo de comportamento também aumentava o poder de influência nas decisões sobre o software:

> Por conta de ser a pessoa que tava alocando parte dos recursos tinha um outro tipo de influencia alem de alguem, um desenvolvedor ativo, além de uma pessoa ativa, um membro ativo. Eu acho que essa questão de estar numa posição de gestão acaba dando esse poder a mais. Mas por outro lado, tudo

> sempre conversado né, e eu diria que eu viabilizava mais, eu entendia bem, até por conhecer tecnicamente o noosfero, eu entendia bem as necessidades do noosfero eu viabilizava que aqueles recursos fossem aplicados nesses gargalos do noosfero, então tinha uma influência, mas geralmente a gente seguia, investia naquilo que o noosfero precisava de fato (Entrevistado 3, Desenvolvedor comum/Articulador de Recursos, UnB, [10951:12759]).

A a primeira hipótese extra que trazemos aqui é, portanto, que o comportamento de Articulador de Recursos pode ser um vetor de aumento da autoridade lateral numa comunidade quando isso é feito por um membro que compartilha os desafios e a identidade da comunidade. Esse comportamento não pode ser encaixado em Trabalho de Coordenação pois a coordenação nesse caso se dá no âmbito do projeto e não da comunidade. Esse tipo de ator, por exemplo, dificilmente assumiria o papel de RM, já que nem commiter era. O fato de esse tipo de ator não fazer parte do grupo de commiters pode indicar os limites dos comportamentos não técnicos para a progressão ao centro da comunidade, mas também pode ser fruto de uma relação mais pragmática: como os projetos baseados em Noosfero articulavam mais de um commiter, essa não era uma necessidade desse tipo de ator.

A segunda hipótese pode ser agrupada numa categoria coringa chamada de contribuições não técnicas. Contribuições não técnicas que não podem ser consideradas como trabalho de coordenação nem articulação de recursos. Nessa categoria coringa aparecem os trabalhos de organização e os trabalhos de gestão de uma instância de Noosfero. O trabalho de organização, que não deve ser confundido com o trabalho de coordenação (pois esse último envolve algum nível de poder e legitimidade para ser desenvolvido), diz respeito à atividades não técnicas voltadas a organizar determinados ativos da comunidade que não o código propriamente dito. Na coleta de dados apareceram trabalhos como produzir a documentação da comunidade (desde funcionalidades do software até diretório de instâncias), reportar falhas no software, testar novas funcionalidades, submeter palestras e artigos para eventos, organizar encontros de disseminação do Noosfero (evangelização). O trabalho de gestão de uma instância de Noosfero é aquele voltado a administrar o serviço viabilizado pelo software, seja uma instância de uma rede de blogs (Blogoosfero), seja um portal dos coletivos de economia solidária (Cirandas.net), seja um portal governamental de participação social (Participa.br). Esse trabalho não implica necessariamente em contribuição de código - ainda que algumas das organizações com instâncias também contribuíam com desenvolvimento - mas é relevante por ser onde a cara pública do noosfero se apresenta para os usuários. Uma instância mal

gerida implica em um serviço ruim e isso afeta a imagem do Noosfero, além da imagem do serviço.

Ainda que esses tipos de trabalho (organização e gestão de instância) não sejam suficientes para que um membro passe a fazer parte do grupo de commiters, já que em última instância o principal objetivo de uma comunidade de software livre é manter uma base de código, ou seja, o software (falaremos disso mais adiante), a barreira de entrada é extremamente baixa:

> Qualquer pessoa [pode fazer um trabalho de organização na comunidade]. Pessoa pode aparecer do nada. Brotou aqui. Eu quero organizar o site do noosfero. Beleza, tá aqui usuário e senha, manda brasa. O que você precisa mais? Se a pessoa não vai onerar a comunidade em termos da comunidade ter que pegar ela pela mão pra fazer, qualquer pessoa pode fazer (Entrevistado 1, Commiter/RM, Colivre, [42644:43160]).

Aqui também, esse caráter aberto é reforçado:

> A parte de comunidade assim, de encontro, de palestra, de conteúdo, isso aí eu acho que é muito de quem tá interessado em fazer, é muito de quem quer puxar mesmo sabe. E aquela coisa, o pessoal fala que a comunidade de software livre tem muitos fazedores né, quem tá a fim de fazer vai lá e faz. Eu acho que quem tá fazendo é quem tem interesse de comunidade pra fazer isso aí, quem tem legitimidade para fazer isso mesmo, pra organizar (Entrevistado 6, Desenvolvedor comum, UnB, [38581:39404]).

Com uma ressalva ao final: “Eu acho que tá certo, tem que ir lá e fazer e se de repente esse cara vai criar uma cultura nova de organização do projeto que ele que puxou depende de ele tá muito tempo na comunidade ou não” (Entrevistado 6, Desenvolvedor comum, UnB, [38581:39404]). Essa ressalva aponta para o que já vínhamos suspeitando, de que o trabalho de organização pode ser um comportamento de aumento de autoridade lateral quando executado de forma iterativa, como um tipo de rotina. Encontramos também alguns indícios de que esse tipo de trabalho aumenta a probabilidade do membro ter seus argumentos considerados e com isso um maior poder de influência desse membro nas decisões da comunidade:

> Se você quisesse uma opinião, não uma opinião porque uma opinião todo mundo pode ter, mas tivesse a opinião ouvida pela comunidade, essa opinião valia mais quando era de uma pessoa que tinha investido tempo e trabalho e código e não só código, mas até mesmo reportando bugs, coisas assim, testando novas funcionalidades. Essa opinião acabava valendo mais, tendo mais peso (Entrevistado 14, Commiter, Colivre, [6279:7134]).

Ainda que seja difícil precisar o quanto esse tipo de trabalho ajuda na progressão do membro em direção ao centro da comunidade (e certamente é cheio de limites), encontramos indícios de que não são somentes as pessoas que contribuem com código que são consideradas importantes para a comunidade: “as pessoas que não estão contribuindo com commits continuam sendo importante né, porque se não tiver gente pra utilizar sem ser os desenvolvedores o negócio vai morrer” (Entrevistado 2, Commiter/RM, Colivre, [10859:11348]). A comunidade se abrir para esse tipo de contribuição, pode ser considerado até como um movimento em direção à uma melhor democracia interna da comunidade, já que o uso e as necessidades das diversas instâncias de Noosfero é que deveriam direcionar o desenvolvimento e não o grupo de desenvolvedores mais ativos, como era de fato (Entrevistado 13, Commiter, Serpro, [6285: 7527]). Por outro lado, numa comunidade eminentemente técnica, essa visão tem limites: “como o noosfero na maioria do perfil das pessoas sempre foi desenvolvedor, é difícil enxergar quem não é desenvolvedor com uma contribuição útil ao projeto, eu acho que os desenvolvedores tendem a achar que útil pro projeto só é código” (Entrevistado 15, Commiter, Serpro, [35095:35543]).

As contribuições não técnicas, portanto, parecem poder ser consideradas como comportamentos de autoridade lateral, pois há evidências fortes de que aumentam a reputação dos seus realizadores na comunidade, porém provavelmente não são suficientes para que o membro atinja posições mais centrais na comunidade. Esses limites, que dizem respeito à valorização de participações não ligadas ao desenvolvimento de código, é talvez um dos principais dilemas de comunidades cuja cultura e lideranças são eminentemente técnicas.

A Tabela 4 a seguir apresenta uma sistematização dos achados em relação às hipóteses complementares de “Articulador de recursos” e “Contribuições não técnicas” e como elas contribuíram para a progressão ao centro da comunidade Noosfero bem como ao acúmulo de autoridade prática.

Hipóteses lançadas pela pesquisa			
Comportamento	Breve descrição	Força na comunidade Noosfero (Autoridade Prática de Abers & Keck, 2013)	Limites percebidos
Articulador de recursos	Faz captação e coordena projetos que fazem uso do Noosfero, investindo na comunidade recursos financeiros e de formação de novos desenvolvedores	Forte: Ganha respeito dos outros membros e capacidade de influenciar nas decisões da comunidade	Ainda que seja suficiente para alcançar a área central da comunidade, não é suficiente para tornar alguém commiter sempre exigindo alianças para ter uma contribuição desenvolvida e incorporada
Contribuições não técnicas	Realiza trabalhos de organização (exceto coordenação) como documentação, testes de versão e evangelização	Fraca: Há um respeito por quem exerce esse comportamento mas não é suficiente para que ele sozinho influencie na progressão ao centro	Não é suficiente para se tornar commiter ou alcançar a área central da comunidade

Tabela 4: Hipóteses complementares de comportamentos para progressão ao centro da comunidade e acúmulo de autoridade prática.

Fonte: Elaboração própria

4.3. Os estrategistas e suas artimanhas

Os comportamentos para progressão de autoridade lateral e acúmulo de autoridade prática ajudam a contar apenas uma parte da história da comunidade Noosfero. Observar como os atores agem e que táticas utilizam para influenciar na decisão sobre o software de uma comunidade vai nos ajudar a complementar esse panorama que começamos a traçar.

Nessa seção vamos analisar as diversas estratégias utilizadas pelos desenvolvedores comuns e líderes da comunidade Noosfero, com vistas a influenciar as decisões sobre o software.

Como já vimos em TSAY, DABBISH & HERBSLEB (2014), as táticas dos membros para influenciar no processo de decisão de uma comunidade FLOSS podem ser descritas de pelo menos quatro formas:

- Pressão na audiência: tática que consiste em desenvolvedores periféricos angariar apoio de outros desenvolvedores também periféricos, além de empresas e organizações para pressionar os membros centrais com objetivo de incorporar suas contribuições;
- Suporte da comunidade: um subtipo da tática anterior, que consiste em utilizar os mecanismos das plataformas de codificação social para conseguir o suporte da comunidade, seja comentando ou marcando as contribuições de modo a indicar que elas são relevantes para uma grande quantidade de membros;
- Suporte do projeto ou empresa: também um subtipo da tática de pressão, que consiste em usar argumentos ligados a um grande projeto ou empresa na tentativa de sensibilizar os commiters na priorização. Sendo uma contribuição que afetaria a adoção do software por uma grande empresa ou projeto, isso certamente conta na priorização do trabalho voluntário que os commiters dedicam para a revisão e incoporação de código;
- Alertando o core: Quando a tática envolve a escolha de um membro específico do core de modo a convencê-lo da importância daquela contribuição.

Ainda que as táticas dos membros são mais explícitas no seu objetivo de influenciar, vamos incluir aqui um conjunto de táticas mais sutis utilizadas pelos líderes que acabam tendo como resultado uma maior porosidade às demandas dos membros que não fazem parte do core da comunidade. Essa tipologia tiramos da abordagem de NAKAKOJI, YAMAMOTO & NISHINAKA (2002), já mencionada aqui:

- Encorajar os membros: Criar uma cultura que promove o pertencimento na comunidade e encorajar os novos membros a se moverem em direção ao centro da comunidade FLOSS através de contribuição contínua;
- Evitar a fragmentação: para evitar a fragmentação (fork) os membros centrais se adaptam e respondem às necessidades e atitudes dos membros da comunidade e estabelecem regras claras para um desenvolvedor periférico se tornar um commiter;

Através do método “descriptive coding” (já descrito aqui) buscamos nos dados indícios de que os desenvolvedores comuns e commiters da comunidade Noosfero lançaram mão dessas táticas. Além de analisar como essas táticas foram usadas, também sugerimos algumas novas hipóteses a partir da descoberta de novas táticas que não foram cobertas pela tipologia obtida em TSAY, DABBISH & HERBSLEB (2014) e NAKAKOJI, YAMAMOTO & NISHINAKA (2002). Além disso buscamos identificar situações em que os atores agiram de forma criativa buscando potencializar o efeito da tática, no que denominamos de componentes criativos. Os componentes criativos podem ser considerados como práticas de construção institucional (ABERS & KECK, 2013) pois inovam na combinação de elementos com o objetivo de alterar as instituições na direção desejada pelos atores. Demonstraremos essas hipóteses a seguir.

4.3.1. Táticas dos desenvolvedores comuns

Nessa seção analisaremos as táticas utilizadas pelos membros desenvolvedores comuns da comunidade Noosfero para ter suas contribuições aceitas na raiz principal do software. As táticas utilizadas pelos desenvolvedores comuns tendem a ser direcionadas aos commiters da comunidade, conforme teoria apresentada pelas autoras TSAY, DABBISH & HERBSLEB (2014) e apresentadas logo acima. Da mesma forma, influenciar a decisão de um commiter requer criatividade política, já que nem todas as regras estão dadas. Dessa forma vamos analisar como os autores utilizaram de forma criativa as práticas de construção institucional para ter sucesso na obtenção de um certo nível de autoridade prática (ABERS & KECK, 2013).

4.3.1.1. Pressão na audiência

O perfil da comunidade Noosfero, como já vimos, é de poucos desenvolvedores ativos, em comparação com grandes comunidades de centenas e até milhares de desenvolvedores e além disso, é uma comunidade com organicidade fraca, já que é formada por organizações autônomas que se relacionavam em torno de um Comum de software. Por

conta disso, seria esperado que o tipo de pressão pública teorizado pelos autores fosse mais raro de acontecer. E foi isso que vimos na análise dos dados:

> Nunca tinha pressão não. Acho que pela própria forma que a comunidade que ela se dá né. Aquilo que eu coloquei que ela nunca teve uma, digamos assim, uma estrutura de comunidade autônoma realmente. Então, todo o trabalho que é feito na comunidade, que não seja proveniente de investimento de alguns dos agentes, ele é assim específico um trabalho de um cliente (Entrevistado 4, Commiter/RM, Colivre, [34037:37115]).

Além de ter sido mais raro esse tipo de tática, ela ficou concentrada no momento em que ainda não havia uma participação maior das outras organizações da comunidade - além da Colivre - no grupo central de commiters. Nessa fase, houve casos de envio de mensagens com cobranças na lista de emails da comunidade - que era o canal em que todos os desenvolvedores participavam - direcionadas principalmente à Colivre, que era a organização que concentrava o papel de incorporar código na raiz principal. Ao relatar esse tipo de pressão, os membros apontam duas abordagens diferentes, uma mais agressiva - que é a que cabe aqui na tática de pressão na audiência - e outra mais sutil que vamos classificar mais adiante na tática "alertando o core":

> Mas tinha gente que mandava na lista, exigindo, que aí normalmente os que mandavam na lista tinha um outro perfil, uma outra forma de falar. No privado era um "na moral aí", na lista era "que merda, já tem tanto tempo que meu negócio tá lá e não tá feito!". Obviamente você recebe de forma diferente cada pedido, uma coisa é pedir com jeitinho outra é pedir com jeito bruto (Entrevistado 2, Commiter/RM, Colivre, [29886:30252]).

A pressa - e a consequente angústia - em ter rapidamente sua contribuição aceita podia ser gerada tanto pela demora em ter a nova funcionalidade ou correção disponível para os usuários (quando a Colivre, principalmente na primeira fase da comunidade, administrava algumas das instâncias de Noosfero) ou pelo fato do desenvolvedor já ter feito uma cópia do código (fork), incorporado e já estar utilizando a sua atualização no seu projeto, mas a árvore principal ainda não refletir essa atualização. Como o acesso ao código é ilimitado, qualquer desenvolvedor poderia fazer o que quiser dentro do seu projeto, sem precisar pedir permissão para o grupo de commiters. Ainda que nesse caso a demora não impedisse o desenvolvedor de entregar a solução para os seus usuários, por outro lado gerava a angústia de estar com uma versão diferente do resto da comunidade. Isso é angustiante pois significa trabalhar com a certeza de, num futuro próximo, ter de enfrentar uma enorme dificuldade para ressincronizar

as bases de código. Esse dilema é bem explicado nessa saborosa explicação de um dos desenvolvedores do Noosfero:

> A ideia positiva do fork é que essa divergência, ela é temporária e rápida né, a medida que passa a integrar código novo. Então, a gente sabia que sempre ia ter um atraso né, o nosso código ia sempre tá divergindo, mas que a gente sempre poderia diminuir ou minimizar esse atraso. [...] Porque o que acontece é que se você começa a atrasar muito, dois, três meses, aí isso causa conflito. Isso gera uma angústia gigantesca no desenvolvedor. É, ele se sente que se separou, sabe! Uma separação assim de um homem e de uma mulher e agora ele não tem mais como conversar. O código, ele não se junta mais. É como se você não pudesse mais transar com a pessoa porque agora você não tem um mínimo de diálogo com ela, sabe! Não tem um mínimo de química com ela, então, é isso, você não pode ficar muito longe. Você tem de manter essa distância controlada, sabe! E quando você não tem acesso ao código, a separação é iminente. [Quando eu mantinha uma instância de Noosfero e ainda não era commiter] era tão dramático que você desfocava né, você começava a desenvolver menos, você começava a ficar preocupado na hora de desenvolver: "Por que eu tô fazendo isso, mas esse código eu não tenho perspectiva nenhuma dele entrar. [...]". Entendeu? Na hora, "Esse código aqui não vai entrar mesmo", você já perde as esperanças. Então, por isso é que a gente fazia a briga política, para não perder as esperanças assim, sabe, no próprio Noosfero, no próprio código que a gente tava fazendo. Porque se a gente não brigasse politicamente para que um dia esse código voltasse, o próprio desenvolvimento do Noosfero perdia sentido, sabe! Você nem tava mais no Noosfero, você tava em outro software que você tinha até que dar outro nome pra ele, sabe. [...] Porque já não é sincronizado mais, você tem de dar um outro nome, sabe! (Entrevistado 5, Commiter, EITA, [31335:34321])

Os membros que optaram por fazer um fork do código, também abordavam de duas formas, uma pressionando e outra tentando sensibilizar de forma mais sutil a comunidade e os commiters da importância de incorporar aquela atualização:

> Porque por exemplo quando alguém contribuiu com alguma coisa e não foi pro master, só que a pessoa já lançou a versão dele com fork, e aí a tentativa normalmente era que o fork não ficasse muito forkado, que a gente não se afastasse muito do principal. Por exemplo, estou lembrando de dois casos, de dois grupos, um eles faziam essa pressão, até pública, de um jeito complicado, como se quisesse arranjar confusão, querendo incomodar mesmo. Enquanto que tinha um outro que fazia, olha eu to fazendo aqui, se vcs quiserem tá aqui depois eu me responsabilizo. Que eu achava até mais interessante, porque você não tem como obrigar alguém a fazer alguma coisa porque não é seu funcionário. Então teve as duas abordagens (Entrevistado 2, Commiter/RM, Colivre, [31266:32106]).

No geral, salvo alguns incidentes dessa primeira fase da comunidade, os membros não apontaram o uso da tática da pressão na audiência como uma prática comum da comunidade

Noosfero. Os caminhos escolhidos tendiam a ser outros, mais sutis ou de articulação política com os membros, como veremos a seguir.

4.3.1.2. Alertando o core

Não encontramos evidências que suportem a tese de que as táticas de "suporte da comunidade" e "suporte de projeto/empresa" tenham sido utilizadas na comunidade Noosfero. Pelo menos não da forma descrita pelas autoras (TSAY, DABBISH & HERBSLEB, 2014), quando um desenvolvedor busca por apoios numa audiência grande que se manifesta pela própria plataforma de codificação social. O Noosfero nunca teve massa crítica para isso, nem de número de desenvolvedores nem de organicidade. E como vimos, as organizações que faziam parte da comunidade estavam todas utilizando o Noosfero para seus próprios projetos (variando de redes de economia solidária até plataformas de participação social), de forma que o argumento de projeto não traria nenhuma distinção. Por outro lado notamos que alertar e sensibilizar um membro específico do grupo de commiters era a tática mais utilizada. Essa tática era utilizada por quase todos os desenvolvedores, buscando dialogar com os commiters muitas vezes até antes de construir o código. Um alinhamento que aumentava muito a probabilidade de ter sua contribuição aceita sem ruídos. Essa tática, dentre as opções trazidas pelas autoras, se encaixa mais no tipo "Alertando o core", porém com dois componentes criativos adicionais que estamos definindo aqui como "Relevância estratégica" e "Vetor de projeto". Esses componentes podem não se apresentar como novas táticas mas como componentes criativos utilizamos dentro das táticas. A ideia de componente criativo aqui, remete a própria ideia de criatividade política, quando atores buscam recombinar recursos existentes (experimentação) e ao mesmo tempo engajam e conectam com outros atores na busca de aumentar seu poder de influência. Grande parte das vezes, o alerta ao "core" acontecia evidenciando a importância estratégica daquela contribuição para a comunidade. Em outras palavras, o desenvolvedor que conseguisse mostrar que aquela contribuição não resolvia apenas o problema do seu projeto ou organização mas sim que melhorava o software como um todo, que trazia um ganho para a comunidade, tinha mais chances de conseguir mobilizar um commiter para a revisão e incorporação do seu código. A isso estamos estamos definindo como um componente de "Relevância estratégica" que aumenta a criatividade da

tática “Alertando o core”. Há indícios de que no processo de socialização da comunidade, interagir e conhecer, se possível pessoalmente, os commiters e usar essa relação para antecipar discussões sobre a melhor maneira de desenvolver uma funcionalidade genérica e relevante para o Noosfero como um todo - e que ao mesmo atendesse a necessidade específica daquele desenvolvedor - era uma tática bastante efetiva para ter suas contribuições aceitas:

> Então, quando eu conheci pessoalmente [os commiters], sempre era legal. Muito bom assim. Até porque um contato pessoal faz você conhecer quem são as pessoas por trás ali. Tem outro desenvolvimento de outras coisas assim, não técnicas né e acaba fazendo você se pertencer àquela comunidade e você entender mais que aquele projeto é relevante, quem são as pessoas envolvidas e etc. Isso era uma coisa que me motivava muito assim e eu sempre aprendi muito assim com as pessoas. Testei bastante, então qualquer dificuldade que eu tive de parear, na hora que eu vi que essas pessoas estavam dedicando tempo pra revisar meu código, mesmo quando não tavam sendo remuneradas pra isso né. Ou quando eu precisava de uma reunião e o pessoal dizia: “Ah, vamu fazer essa reunião aí, vamu trocar essa ideia”, e ver que esse pessoal tava disponível, foi muito legal (Entrevistado 6, Desenvolvedor comum, UnB, [42888:43984]).

A não ser que já houvesse algum contato prévio à realização da contribuição, geralmente as submissões de código não eram marcadas para nenhum commiter específico:

> Geralmente cabe mais ao commiter ir no repositório e ver que contribuições estão pendentes né. Se tiver alguma coisa que interessa ser revisado ou que você tem capacidade de revisar, você vai lá e pega aquela minha “pull request” [i.e. submissão de código] pra você né, pra você tocar. Geralmente não têm solicitação direta (Entrevistado 9, Commiter, UnB, [30730:31224]).

Dessa forma, o recurso de menção (“@”) na plataforma de codificação social não era usado como um meio de sensibilização em si, mas para dar consequência a conversas já iniciadas:

> Normalmente [quando você é marcado diretamente] é alguma [submissão] que você já conversou antes, né, por exemplo, com alguma pessoa que você já tava em contato antes. E ela já tava falando sobre a contribuição de alguma forma né e que tinha interesse em fazer aquilo e perguntando qual a melhor forma. Então, meio que já existe um contato prévio e a pessoa abre uma g-quest e geralmente te menciona no comentário né. Comentá lá: “Gabriel, você pode revisar isso aqui? Terminei!”. Aí de uma g-quest disponível, a gente vai lá, pega e revisa. Geralmente é nesse sentido (Entrevistado 9, Commiter, UnB, [31239:32018]).

Os desenvolvedores mais integrados na comunidade (não necessariamente commiters) tinham bastante consciência da importância desse componente:

> O jeito mais fácil [pra submeter contribuições] era chamar pra parear e definir, aí a gente tinha o caminho das pedras e depois submetia e tinha o

> processo normal de revisão. Vou dizer que 90% das contribuições que fiz eu tive contato direto inclusive pareando com alguns dos commiters da época. [...] Mas tinha uma influência do papel que eu tinha, eu acabava sendo privilegiado nesse suporte nessa atenção, porque eu tava tratando ali interesses além daquela contribuição [interesses da comunidade] né, então eu acabava tando numa posição privilegiada para pedir esse tipo de ajuda (Entrevistado 3, Desenvolvedor comum/Articulador de recursos, USP e UnB, [17846:18891]).

Do lado dos commiters, a motivação em contribuir aumentava quando ficava mais visível o objetivo comum de evoluir o software e não apenas a evolução de um caso de uso específico: "eu me sensibilizava por perceber que o objetivo do grupo era parecido com o meu, de querer que o software todo fosse evoluido, não só o dele, não só o objetivo dele" (Entrevistado 2, Commiter/RM, Colivre, [37288:39088]). Aparece também uma ideia de que havia um jeito certo de contribuir, e esse jeito era pensar no software como um todo:

> A gente tava tinha muita contribuição de fora da colivre, e aí a ideia de querer manter as pessoas contribuindo, mas contribuindo do jeito certo. contribuindo de forma que a contribuição atingisse a maior parte de pessoas, fosse bom pro software em si e não pra versão que a pessoa está utilizando (Entrevistado 2, Commiter/RM, Colivre, [52270:53955]).

Da mesma forma, o trabalho voluntário de revisão, "um favor", fazia mais sentido se fosse pra algo relevante para a comunidade:

> Ninguém tá pagando a gente assim pra poder tá fazendo isso né, então… Então, nunca foi, nesse ponto de vista de pressão, sempre foi uma coisa mais de favor, né. Favor, considerando que aquilo é uma coisa importante, geralmente a pessoa saberia que eu ia avaliar como uma coisa relevante também de eu me envolver, ela ia me acabar pedindo e eu me envolvia (Entrevistado 4, Commiter/RM, Colivre, [34037:37115]).

Como um espécie de potencializador, que aqui chamamos de componente criativo, a conversa iniciada com o mote "isso faz sentido pro Noosfero" faz com que o processo de revisão e incorporação seja muito mais fluido:

> Geralmente, quando é uma coisa mais complexa né, mais específica, perguntar o que que o core commiter acha, se faz sentido mandar aquilo pro Noosfero, depois fazer. Geralmente, quem tá começando a contribuir agora faz isso né. Por exemplo, o pessoal da UnB, sempre faz isso, sempre tá em algum canal de comunicação ou pergunta diretamente se aquilo faz sentido, se é um bug mesmo, se faz sentido eles abrirem um "merge request" [submissão de código] para aquilo. A gente fala que sim, toca a conversa daí, depois eles abrem e acaba acontecendo [...] deles direcionarem no merge request pra dar uma olhada já que a gente já começou a conversar (Entrevistado 9, Commiter, UnB, [38674:40366])

Há fortes indícios portanto de que a tática “alertando o core” foi a tática mais utilizada na comunidade noosfero, incluindo os dois componentes criativos que identificamos nos dados, o “Relevância estratégica” já apresentado aqui e o “Vetor de projeto” que falaremos a seguir.

O componente criativo “Vetor de Projeto” tem relevância para o caso estudado pois foi utilizado por boa parte das organizações que tiveram força de desenvolvimento própria de Noosfero para criar uma dinâmica permanente de dedicação dos principais commiters enquanto havia projetos em curso. Como o próprio nome já diz, esse componente diz respeito a usar recursos de um projeto envolvendo Noosfero para convidar commiters a fazerem parte da equipe do projeto com o objetivo de orientar o time local nas principais tarefas de desenvolvimento, garantindo que esse trabalho seguisse o ritmo do projeto e que as contribuições fossem feitas de forma compatível com o Noosfero e não algo específico do projeto. Entre as organizações que utilizaram desse componente, estão a Cooperativa EITA (na primeira fase do projeto), a USP, UnB e o Serpro, esse último através dos orgãos finalísticos que estavam utilizando o Noosfero em seus projetos como a Secretaria-Geral da Presidência da República (Participa.br) e a Secretaria Nacional da Juventude (Portal da Juventude).

O componente criativo “Vetor de Projeto” funcionava através da criação de um fluxo de desenvolvimento que envolvia diretamente um ou mais commiters da comunidade. Era esperado desses commiters, nesse caso contratualmente, que eles orientassem e revisassem as contribuições que eram feitas no âmbito do projeto:

> [No projeto] a gente sempre manteve aquele fluxo de alguém desenvolver, alguém do projeto já dava uma revisada então a gente já tinha o feedback antes. No geral a gente mandava um PR [submissão de código], a gente pedia alguém específico que tava geralmente associado ao projeto porque o projeto que eu tava a gente tinha vários commiters trabalhando no mesmo projeto (Entrevistado 12, Commiter, UnB, [22939:23691]).

Nesses casos, confirmando essa tática como de tipo “Alertando o core”, a comunicação era feita diretamente com os commiters alocados para o projeto:

> Não era uma coisa assim de mandar na lista [geral] lá, era uma coisa de conversar com as três pessoas ali mais ativamente mas que também pertencia ao nosso projeto, a gente já falava assim: “Ó, vai precisar fazer isso, isso e aquilo”. Já trocava ideia ali com aquelas pessoas e pronto, entendeu? Já era um coisa mais assim. Nunca fomos de discutir muito na lista [geral]. [Essas

> pessoas que a gente acionava] eram commiters né, [...] que eventualmente participaram do SPB [Projeto envolvendo noosfero] né em algum período. Então, a gente já trocava ideia com elas ali mesmo durante os hangouts e tal. Não era uma coisa que a gente ficava muito expondo em lista. [...] Eles tinham responsabilidade de participar das reuniões que a gente fazia do projeto. [...] Geralmente uma delas ficava responsável por revisar sim. Que usavam meio que o slot do projeto delas pra revisar código, muitas vezes (Entrevistado 6, Desenvolvedor comum, UnB, [23600:26145]).

E o papel desses commiters alocados para o projeto era importante para que as contribuições entrassem no padrão Noosfero e fossem incorporadas na raiz principal como um ativo para toda a comunidade:

> Tinha uma pessoa determinada já pra gente ali. Era o commiter que tivesse participando do nosso projeto ali na USP naquele momento. [...] Quem tava cuidando do projeto na USP, eu acho que fechava o contrato na Colivre também de consultoria. Aí ficava uma pessoa ali alocada. [...] Por exemplo, ah a gente tinha uma dúvida fazer uma mudança, a gente perguntava qual que era a melhor forma e ele já explicava pra gente. E seguindo isso o time já conseguia ficar mais aderente ao padrão né (Entrevistado 8, Desenvolvedor comum, USP, [24145:25195]).

Os arranjos eram flexíveis, havendo a possibilidade do commiter dedicar algumas horas de revisão para o projeto e essas horas serem remuneradas por contratos com a Cooperativa Colivre:

> Por exemplo, um código que eu consegui incorporar e que tinha recurso na época foi o plugin de buscas pelo Apache Solar né, que é uma engine em buscas, é uma base de dados de buscas desenvolvida pelo Apache, pela fundação Apache. E eu tive reuniões e reuniões e mais reuniões com [a commiter] e o tempo dela foi pago pelo projeto. [...] [O trabalho não foi de mobilizar o commiter] era pagar mesmo. Era diferente a lógica. É pagar a Colivre: "Óh, temos esse código aqui, a gente precisa integrar ele. Então, a gente vai pagar as horas que forem necessárias pra vocês fazerem esse processo aí junto com a gente". A ideia era minimizar claro, né, a parte da Colivre e a gente fazer o máximo possível pra diminuir esse custo né, que a hora da Colivre era mais cara (Entrevistado 5, Commiter, EITA, [53070:55060]).

Outra forma de utilizar o componente criativo "Vetor de projeto" era a organização trabalhar para ter os seus próprios commiters. Todas as organizações ativas no desenvolvimento do Noosfero acabaram chegando a esse ponto em algum momento. Na rotina diária de trabalho, a tática continua a mesma, ou seja, a organização depende desse

commiter, alguém do core, para ter seu código incorporado, a diferença é que, em vez de ser alguém puxado de fora pra dentro, seria alguém revelado pelo própria organização.

Em um determinado momento da história do Noosfero, as organizações que tinham equipes de desenvolvedores começaram a lutar politicamente para que o círculo de commiters fosse aberto. Como já demonstramos, num primeiro momento essa ideia era recebida com um certo receio pela organização principal, a Colivre, pois sua sustentabilidade já estava, àquela altura, 100% vinculada ao Noosfero. Por outro lado, havia um interesse da própria Colivre na aproximação dessas organizações, para que novas contribuições pudessem ser feitas de maneira mais ágil e o software não ficasse totalmente dependente deles. Também por uma questão econômica, pois num determinado momento, havia mais recursos de desenvolvimento vindo de organizações externas do que da própria cooperativa:

> Outra questão foi flexibilizar um pouco essa questão de quem entra né, como commiter do Noosfero. A Colivre estava sempre tentando centralizar, ter o papel de protagonismo e ao mesmo tempo ela começou a entrar em crise econômica. Ela não tinha a maior parte dos editais, outras organizações tinham mais recurso para trabalhar com o Noosfero. [...] A Colivre, então, ela, mais por motivos financeiros do que políticos, ela foi obrigada, pra que o Noosfero continuasse, ela foi obrigada a criar critérios objetivos, simples e fáceis pra que as pessoas possam se tornar commiters (Entrevistado 5, Commiter, EITA, [23278:24943]).

Esse dilema foi resolvido a partir da criação da regra, já apresentada aqui, para um desenvolvedor se tornar um commiter. Além de ter um mínimo de experiência no desenvolvimento do Noosfero, essa pessoa teria que ser endossada por dois commiters atuais. Essa regra permitiu a entrada, do dia pra noite, de um commiter por organização, pois muitos já estavam com experiência acumulada, e a luta política envolveu articulação alguns dos commiters atuais para o endosso. Para essas organizações, a criação dessa regra representou um alívio nos seus fluxos de trabalho. Principalmente para aquelas que tinham dificuldade em contratar commiters durante longos períodos de tempo. Nesse trecho é possível perceber de forma clara essa necessidade:

> Uma das motivações pra gente ter mais commiters lá, era algo desse tipo, porque tinha que ter esse trabalho de convencimento grande e como a gente tava caminhando numa direção, tinha outras organizações que caminhavam com outros objetivos, as vezes as coisas não convergiam e isso pra gente atrapalhava bastante, então esse trabalho de convencimento que tinha que ter foi até reduzido quando a gente teve mais commiters, no caso quando eu virei commiter do noosfero (Entrevistado 13, Commiter, Serpro, [26536:27001]).

Aqui também:

> [Quando eu me tornei commiter] foi um momento que eu tava bastante ativo no noosfero, no contexto de um projeto que a gente participava lá no laboratório lappis da unb, e aí a gente sentiu essa necessidade de ter algum do projeto mais ligado ainda a comunidade, com mais poder vamos dizer assim, pra aceitar as coisas que a gente desenvolvia no projeto, revisar (Entrevistado 7, Commiter, UnB, [15700:16515]).

Em suma, ambas as estratégias tinham como objetivo garantir a atuação de algum commiter no projeto, principalmente para fins de agilidade no cronograma interno do projeto, mas também para garantir que o software desenvolvido entrasse como contribuição no Noosfero, ou seja, continuasse sendo Noosfero. No primeiro caso, o elemento mais forte era o recurso financeiro para contratar algum commiter já existente na comunidade. O segundo era uma estratégia de médio prazo, pois a organização deveria formar seus próprios commiters. Essas duas vertentes do componente criativo "Vetor de Projeto" foram utilizadas pelas organizações parte da comunidade Noosfero.

O que os componentes criativos demonstram é que a tipologia da tática "alertando o core" não deu conta de captar o caso concreto da comunidade Noosfero. Ainda que essa tática tenha sido a mais utilizada pelos membros da comunidade, foi realizada de formas tão variadas que sentimos a necessidade de agrupá-las nesse novo conceito de componente criativo que nos ajudou na explicação do fenômeno. Mesmo com limitações, esses componentes reforçam a ideia de que os membros recombinavam recursos e faziam com que eles fluíssem pela rede da comunidade de forma a garantir a sua influência na incorporação dos códigos relevantes para o seu projeto. No caso do componente criativo "Relevância Estratégica" o desafio era sempre resignificar a necessidade concreta de sua própria organização de uma forma que fosse útil na evolução do conjunto do software, chamando assim a atenção das outras organizações. Formular uma proposta que fosse coerente para essas organizações só seria possível tendo um conhecimento mínimo dos objetivos e casos de uso de cada uma delas, de forma que a conectividade era um elemento fundamental para o sucesso da utilização desse componente criativo. De forma parecida, o componente criativo "Vetor de Projeto" implicava na criação de novos atalhos para as áreas mais centrais da comunidade (seduzindo e contratando commiters para participar dos projetos) enquanto se investia na progressão dos desenvolvedores da própria organização. O contato próximo com

desenvolvedores commiters, principalmente os históricos/fundadores acelerava também o processo de formação desses novos líderes. Dessa forma, estamos considerando que a utilização dos recursos do projeto para esse fim foi uma ação criativa dos membros da comunidade para criar esses atalhos nas regras vigentes daquele momento na instituição comunidade Noosfero. Essa ação criativa também permitiu aumentar os níveis de autoridade prática das organizações que as utilizaram, permitindo que elas obtivessem mais facilmente o resultado de incorporação de código desejado.

Na tabela 5 abaixo, há um resumo analítico das táticas utilizadas pelos desenvolvedores comuns da comunidade Noosfero, assim como o efeito dos componentes criativos na tática "Alertando o core".

Táticas utilizadas pelos DESENVOLVEDORES COMUNS			
Tática	Breve descrição	Ocorrência na comunidade Noosfero	Componente criativo (atuação criativa dos membros, recombinando e criando novas situações de modo a aumentar a autoridade prática da organização e influenciar na decisão sobre incorporação de código)
Pressão na audiência	Tática que consiste em usar a plataforma de codificação social para angariar apoios para pressionar os membros centrais	Muito baixa por ser uma comunidade com baixa escala. Aconteceu pontualmente na primeira fase via lista de emails	Para essa tática não foram encontrados componentes criativos
Suporte da comunidade	Tática que consiste em utilizar a plataforma de codificação social para demonstrar suporte a uma determinada sugestão de código	Não encontramos indícios de que essa tática tenha sido utilizada devido a baixa escala da comunidade	Não foram encontrados componentes criativos para essa tática

Suporte do projeto ou empresa	Tática que consiste em referenciar uma empresa ou projeto como argumento de força na defesa de uma sugestão de código	Não encontramos indícios de utilização, até porque todas as organizações da comunidade estavam relacionada a algum projeto então essa não seria uma distinção relevante para o caso estudado	Não foram encontrados componentes criativos para essa tática
Alertando o core	Tática que consiste em mencionar individualmente um membro do grupo de desenvolvedores centrais de modo a convencê-lo da importância da contribuição	Foi de longe a tática mais utilizada na comunidade Noosfero, onde o grupo de desenvolvedores centrais funcionava como um espécie de colégio de líderes. Em alguns casos as conversas aconteciam até com antecedência à construção do código que seria submetido	Para essa tática foram encontrados dois componentes criativos que culminaram com o aumento do nível de autoridade prática das organizações que os utilizaram: “Relevância estratégica” e “Vetor de projeto”. O primeiro está ligado a construir a submissão de uma forma que tenha relevância estratégica para a comunidade. O segundo está ligado à criar atalhos paras as áreas centrais da comunidade tanto envolvendo commiters nos projetos (aproximar um commiter com compromisso de trabalho), como também em formar novos commiter a partir da organização (prata da casa).

Tabela 5: Resumo das táticas utilizadas pelos desenvolvedores comuns da comunidade Noosfero

Fonte: Produção própria

4.3.2. Táticas dos líderes

Como vimos, as táticas utilizadas pelos membros tinham uma motivação direta em influenciar a decisão de incorporação de códigos na raiz principal da comunidade. Por outro

lado os líderes utilizam certas táticas destinadas a obter benefícios para o software e para a comunidade, que acabam culminando no aumento da porosidade dos processos decisórios da comunidade, pela incorporação de contribuições dos desenvolvedores periféricos. Na teoria estudada (NAKAKOJI, YAMAMOTO & NISHINAKA, 2002) identificamos duas táticas exercidas pelos líderes que são comuns às comunidades de software livre orientadas a serviço (que é o caso do Noosfero), que são "Encorajar membros" a continuar contribuindo e "Evitar a fragmentação" se adequando às necessidades dos membros e criando regras claras de como se tornar desenvolvedor central (commiter).

Encontramos de forma bem explícita na comunidade Noosfero a preocupação com a tática de "Encorajar membros":

> Em qualquer projeto de software livre você quer reduzir o máximo possível o tempo entre a pessoa submeter uma contribuição e ela ser aceita ou rejeitada. [Porque reduzir?] Primeiro pra que a pessoa que ta fazendo a contribuição tenha feedback rápido. Assim, por exemplo, uma pessoa que ta querendo se iniciar pra contribuir com um projeto, é um fator de motivação bem grande a contribuição dela ser aceita. Ficar lá pra sempre, sem resposta, não é o ideal (Entrevistado 11, Commiter/RM, Colivre, [24438:25738]).

Além de não deixar muito tempo sem resposta, havia também uma preocupação em como responder, de forma a manter o interesse daquela pessoa:

> Eu me preocupei muito com isso, com manter as pessoas. por exemplo, você falar não pra uma pessoa em relação a código obviamente, você falar não para uma contribuição, tem que ser um não de um jeito que, "olha eu quero muito que você contribua mas não desse jeito", mas falar de um jeito, eu me preocupava muito em como falar, [...] então eu sempre tentava ser muito simpática e muito gentil - e mesmo assim fui chamada de rainha mandona, imagina - mas eu me preocupava muito. Mas sobre o futuro da comunidade, eu me preocupava mais em manter as pessoas contribuindo e fazer com que o noosfero conseguisse atingir públicos maiores (Entrevistado 2, Commiter/RM, Colivre, [52270:53955]).

Quando uma nova release era publicada, também havia essa preocupação no momento de escrever as notas da release, que era o instrumento de divulgação da nova versão:

> A preocupação era bem essa, em tentar fazer com que as pessoas continuassem envolvidas também. Deveria ter um interesse nisso, fazer com que as partes interessadas, as notas de release tinha a ver com isso porque como a gente agradecia e tal, queira ou não todo mundo tem um pouquinho de vaidade, se você contribui com a coisa, as pessoas querem ser reconhecidas por aquilo e é importante que elas sejam reconhecidas por aquilo (Entrevistado 2, Commiter/RM, Colivre [46760:48421].

Ao mesmo tempo, embora essa preocupação em “Encorajar membros” fosse praticamente consensual, a postura das lideranças mais fortes do momento influenciavam muito como essa abertura seria percebida pelos desenvolvedores periféricos. Como vimos, a liderança do noosfero não era exercida necessariamente pelo RM e sim pelo commiter que pelos comportamentos de “contribuições técnicas” e “comunicação técnica” progredia lateralmente e atingia os maiores níveis de autoridade prática. Quando essa liderança tinha uma postura mais fechada no diálogo com as contribuições externas, isso afetava e era sentido pela comunidade:

> Eu vejo que a questão da liderança pra um projeto de software livre foi crucial. Hoje eu vejo o Noosfero sendo liderado por pessoas que conseguem dialogar e que têm muita possibilidade de diálogo com os outros desenvolvedores e com as diferentes contribuições. Porque na época o Noosfero era muito fechado. Ele tinha muita dificuldade de dialogar com contribuições externas, praticamente nada fora da Colivre entrava no código do Noosfero. Eu fui uma das primeiras pessoas que tava cobrando que isso fosse possível acontecer na prática. Então, eu tive muito choque com a comunidade pra poder abrir ela e fazer com que ela se tornasse mais flexível e não, às vezes por se mostrar muito pedante, muito rígida, com um critério de qualidade muito alto que ninguém conseguiria contribuir código né. Tem que ter uma certa flexibilidade pra que as coisas possam andar e sejam melhoradas com o tempo e não exigir perfeição. Existia um certo perfeccionismo com uma coisa muito ideal que ninguém conseguia alcançar (Entrevistado 5, Commiter, EITA, [6535:8004]).

Em suma, há fortes indícios que a tática de “Encorajar membros” exercida pelos líderes da comunidade Noosfero, oscilava entre uma preocupação em tratar bem e não deixar ninguém sem resposta, mas ao mesmo tempo com uma dificuldade em abrir mão de certos perfeccionismos técnicos que desmotivavam alguns contribuidores engajados. Tudo indica que esse é um equilíbrio difícil de obter, pois baixar muito a régua da qualidade, ainda que seja saudável para a vida social da comunidade, também pode inviabilizar a comunidade no longo prazo, envenenando a “galinha dos ovos de ouro” que sustenta a comunidade, ou seja, o software. Como o foco da nossa análise aqui não é a qualidade do código ou do software, mas a relação entre os membros no contexto do exercício dessa tática específica, podemos perceber que o discurso técnico utilizado para recusar contribuições parecia não estar ancorado em uma base comum à toda a comunidade. No caso do Noosfero essa base parecia existir apenas para os contribuidores que faziam parte da organização principal, a Colivre. Em outras palavras, para os de dentro, as justificativas para a recusa eram razoáveis. Para os de

fora, soava como algo pessoal, “era questão ideológica não era de código” (Entrevistado 15, Commiter, Serpro, [30936:32029]), ambas comunicadas como uma questão de “política técnica”:

> Eram sempre discussões muito políticas, técnico-políticas né, porque você estava sempre ali falando: “Ah o código desse jeito não pode porque a política de código, ou seja, porque a política técnica nossa não permite um código desse”. Então, imagina duas coisas juntas, política já complica tudo (Entrevistado 5, Commiter, EITA, [10591:11681]).

A tática de “Evitar a fragmentação” na qual os líderes se adequam às necessidades dos membros também foi bastante utilizada na comunidade Noosfero. Essa preocupação está muito presente na narrativa dos membros e diversos exemplos concretos demonstram como isso não era um discurso vazio, mas sim uma prática da comunidade, especialmente da organização principal, a Colivre. O fato da cooperativa ser oriunda de dois movimentos sociais - economia solidária e software livre - fortalecia a prática de ser uma organização comunitária que olhasse de forma generosa para toda a comunidade. Para os adeptos do FLOSS, a ocorrência de uma fragmentação por motivos menores seria um fracasso de ação coletiva do grupo e só levaria ao enfraquecimento de toda a comunidade: “[Eu fui contra o fork] porque a comunidade já era fraca, pequena, você dividir o que é pequeno não tinha sentido nisso. E na minha visão [absorver as diversas necessidades] é um jogo de ganha ganha” (Entrevistado 15, Commiter, Serpro, [11770:12436]). Além disso havia também a motivação econômica, pois a sustentabilidade financeira da cooperativa ficava mais resguardada se as mudanças não fossem muito bruscas, ou seja, não afetassem os casos de uso instalados ou, em outras palavras, não atrapalhassem o os contratos vigentes com os clientes da cooperativa. Esses dois componentes, a preocupação do movimento em não fragmentar forças e os incentivos econômicos fazia com que fosse muito presente na prática dos líderes esse instinto de adequação às necessidades dos membros e organizações da comunidade.

Um exemplo concreto disso vem da prática de flexibilizar o lançamento das versões caso houvesse demanda de algum membro. Durante um bom tempo o Noosfero trabalhou com a lógica de congelar as contribuições dias antes do lançamento de uma nova versão, para que o RM pudesse trabalhar nos testes e confecção das notas da versão. Por mais que os lançamentos de versão sejam momentos críticos para um software, os líderes buscavam adequar esse momento ao ritmo dos outros desenvolvedores:

> Então, tinha um momento que eu como release manager eu falava: “Ó galera, daqui a duas semanas tô planejando aqui pra fazer freeze pra poder fazer a próxima versão”. E esse freeze, durante esse momento não podia entrar nada na árvore. Era um momento de a gente meio que consolidar ali, só poderia entrar correção de problemas. Enquanto consolidasse aquela versão ali, a gente lançava depois. Isso poderia durar uma semana. Então, em alguns momentos algumas pessoas poderiam vir falar: “Pô, tem como adiar o freese sei lá mais uma semana porque eu quero que isso entre na versão”, sacô? [...] Geralmente eu aceitava (Entrevistado 4, Commiter/RM, Colivre, [57650:61302])

Outro exemplo dessa preocupação acontece quando o Noosfero fez o lançamento de uma versão que trazia um componente - no caso a interface para celulares - que era radicalmente diferente da versão anterior. Isso significa que para as organizações que mantinham instâncias de Noosfero a atualização para a nova versão não era possível sem fazer um trabalho de adequação técnica. Como nem todas as organizações tinham condições de fazer isso no curto prazo, elas perderiam também os outros recursos importantes que estavam sendo publicados naquela mesma versão. Caso os líderes do Noosfero não olhassem e buscassem uma solução para essa questão, o risco de fragmentação seria muito alto: as pessoas seguiriam usando a versão anterior e todo novo desenvolvimento feito por essas organizações seria com base na versão antiga, perdendo o sincronismo com o Noosfero oficial. Nasceria então uma espécie de “Noosfero velho” com o risco de nunca mais voltar a conversar com a versão oficial, fragmentando esforços e rachando a comunidade. Sensíveis a essa questão e utilizando a tática de se adequar a necessidade dos membros para evitar a fragmentação, os líderes do Noosfero tomaram a decisão de publicar duas novas versões oficiais: uma com o componente novo de celular, outra com o componente velho, mas ambas atualizadas em relação aos outros recursos. Com isso, as organizações tiveram um prazo bem alargado para fazer as adequações técnicas necessárias para receber o novo componente de celular, ao mesmo tempo que todas as suas contribuições ao código Noosfero seriam incorporadas na raiz principal do software e não seriam perdidas. Com isso, a comunidade seguiria unida trabalhando com um Noosfero só:

> E aí a gente conseguiu, depois de muitos meses de trabalho e tudo mais chegar numa versão estável dessa nova interface já responsiva. E aí como mexer na fatoração dessa interface quebraria basicamente com todos os Noosferos instalados no momento né porque essa migração não era retroafirmativa [ou seja compatíveis] com os temas que tinham sido implementados com a interface anterior, aí eu meio que tive de tomar a decisão de seguir duas versões independentes do Noosfero. Então, o Noosfero hoje tem a versão 2.x e a versão 1.x que tão seguindo

> independentes. A ideia é que a 1.x ainda tá na interface antiga e algumas coisas devem ser implementadas, correções de problemas e algumas funcionalidades também, a gente porta da 2 pra 1 porque muitas pessoas ainda tão usando (Entrevistado 4, Commiter/RM, Colivre, [64494:67524]).

Um último exemplo sobre o uso da tática de adequação aos membros para evitar a fragmentação foi quando um grupo da UnB se interessou em trabalhar no aprimoramento do sistema de permissões do Noosfero. O sistema de permissões era responsável por fazer funcionar as opções de permissão sempre que algum usuário criava um conteúdo novo na rede. Quando você escreve um post no Noosfero, assim como no Facebook, há diversas opções de permissão para esse post, exemplo: post público, somente amigos, escrita permitida para os membros de um grupo específico etc. Porém as possibilidades de escolha de permissões no Noosfero eram muito mais detalhadas, chegando a quase uma dezena de opções. Se por um lado isso amplia os usos possíveis do software, por outro torna essa operação muito mais difícil. Diante desse diagnóstico, um grupo de desenvolvedores do Lappis/UnB encarou a tarefa de simplificar essas permissões em duas ou três opções gerais, para melhorar a usabilidade do Noosfero, ou seja, simplificar seu o uso. A intenção era boa, mas isso iria quebrar diversos usos de outras instâncias e grupos de desenvolvedores que já utilizavam aquela estrutura de permissões da maneira que tinha sido consolidada. Por conta disso o grupo encontrou diversas resistências dos líderes do Noosfero para executar a tarefa da maneira que tinham pensado. Se por um lado isso frustrou a expectativa desse grupo, por outro evitou o trauma de impor uma alteração para os outros grupos que estavam desenvolvendo e utilizando Noosfero. Esse exemplo é importante para se perceber que nem sempre a tática de adequação aos membros se dá aceitando uma nova contribuição. Algumas vezes ela pode se dar recusando ou ajustando a contribuição de um grupo específico para evitar gerar traumas para outros grupos e consequentemente a fragmentação da comunidade em bases de código diferentes:

> Ou seja, se você tivesse alguns perfis, alguns tipos de perfis que já tivessem regras [de permissão] pré definidas e não precisasse ficar verificando o tempo todo as sub regras do conteúdo, seria muito mais fácil em termos de código né. Mas como tinha essa complexidade: “Ah, os projetos já estão instalados, eles têm um perfil assim já, tá tudo aberto e vai deixar de tá aberto, as pessoas vão se confundir e tal”, tinha que deixar flexível a ponto de: “ah, tudo vira uma opção né”. Então, você vai editar um artigo lá no Noosfero, você tem milhões de opções de privacidade lá pra você permitir ou não [deixando super confuso] (Entrevistado 6, Desenvolvedor comum, UnB, [31256:32801]).

Passamos aqui por três casos concretos relacionados ao uso da tática de adequação aos membros para evitar fragmentação. O primeiro sobre os líderes flexibilizando a data de lançamentos das versões, outra com os líderes assumindo o ônus de suavizar o processo de atualização dos outros grupos de desenvolvedores e finalmente uma sobre rejeição de contribuições que seriam boas para o software mas que causariam traumas nos outros grupos e aumentariam o risco de fragmentação. Esses três exemplos têm em comum o fato de todas serem decisões que foram tomadas pelas lideranças quando confrontadas com a escolha entre levar em consideração as necessidades dos outros membros ou não. Porém no processo de análise dos dados, percebemos um componente criativo para essa tática que consiste em utilizar uma abordagem técnica embutida no próprio software para conferir maior autonomia aos diversos grupos desenvolvedores, ao mesmo tempo neutralizando o impacto causado por lideranças inflexíveis. No caso do Noosfero, essa abordagem técnica consistiu na construção de uma infraestrutura de plugins, com o objetivo explícito de diminuir os requisitos de revisão e aumentar a agilidade com que o código externo era aceito. Já explicamos que a infraestrutura de plugins foi também um componente tecnopolítico fundamental para a façanha da comunidade de ter permanecido unida. Aqui, ela desempenha um papel de oferecer instrumentos para que as lideranças tivessem mais possibilidades de se adequar às necessidades dos membros. Isso acontece porque com essa infraestrutura de plugins o Noosfero se transformou numa plataforma que passou a aceitar aplicativos externos sem a necessidade de passar por um criterioso processo de revisão de código:

> Foi uma coisa chave porque permitiu contribuições menos acopladas com o core do projeto, assim por exemplo, o nível de criticidade de entrada de código novo se for via plugin é muito menor do que se for no core. O plugin pode entrar com menos qualidade de código, mas o core não. Então uma contribuição que alguém externo mandava (um não desenvolvedor, um não commiter) se ele mandava uma contribuição pra um plugin o nível de revisão do código era muito mais relaxado. Mas se ele mandava pro core, o nível de revisão era muito mais crítico. O projeto noosfero conseguiu aumentar a vazão de contribuições que recebia, com isso (Entrevistado 1, Commiter/RM, Colivre, [51760:52701]).

A esse componente criativo da tática de adequação aos membros para evitar a fragmentação estamos também utilizando o nome de “Tecnopolítico”. Tecnopolítico porque consiste no desenvolvimento de uma tecnologia voltada a resolver um gargalo político da comunidade

ligado à dificuldade que a Colivre tinha de justificar perante os outros grupos a real necessidade de alguns requisitos de qualidade, que acabavam sendo vistos como inalcançáveis por quem não estava integrado ao núcleo duro da cooperativa. Esse gargalo gerava uma série de tensões:

> Tinha sempre aquele esquema de moderação de código, sabe! Ah, o seu código tem que ser aceito por outra pessoa e eu acredito que isso gera uma desconfiança, é isso! Eu acho que o principal é você não quebrar o código dos outros, mas seu código não precisa tá perfeito pra ser colocado junto da árvore né. Então, nunca deu certo. Enquanto não mudou a liderança do Noosfero, o meu código nunca foi integrado. Eu acho que aí teve uma mudança fundamental aí que é uma mudança de que você confia nas outras pessoas (Entrevistado 5, Commiter, EITA, [11889:14468]).

Esse componente criativo tecnopolítico foi arquitetado pelos líderes como uma alternativa de conciliação, uma forma de mitigar o risco de rompimento de alguns grupos com a comunidade e consequentemente a sua fragmentação. Há fortes indícios de que a estratégia deu resultado pois temos relatos que confirmam que, nos momentos de maior divergência e tensão, o desenvolvedor periférico tomava a decisão de fazer em plugin só pra não ter que seguir disputando com as lideranças (Entrevistado 15, Commiter, Serpro, [30936:32029]). Também identificamos um exemplo onde os próprios líderes solucionaram um conflito utilizando o artifício desse componente criativo tecnopolítico. Foi no episódio da incorporação de um novo mecanismo de busca no Noosfero, momento em que essa nova atualização importante para um dos grupos acabou gerando efeitos negativos no software para os demais grupos. Foi um momento de muita tensão para a comunidade, mas a habilidade das lideranças e a possibilidade de utilizar o componente criativo tecnopolítico foram decisivas para que a situação se resolvesse e não houvesse a fragmentação:

> Depois de ter sido incorporada, essa mudança acabou trazendo várias características negativas do ponto de vista de performance e até de resultado mesmo como funcionalidade de busca né do Noosfero. E aí meio que surgiu essa ideia de a gente, já foi incorporado essa engine de busca e já foi utilizada por alguns agentes, então a gente não pode simplesmente jogar fora e falar assim: “A gente vai voltar pro que era antes”. Então, aí foi meio que motivado por essa questão aí de como a gente pode, digamos assim, achar um meio termo que funcione pra ambas as partes. E aí a gente veio com essa ideia do desenvolvimento da infraestrutura que permitia que quem quisesse usar aquela engine, usasse e quem não quisesse pudesse seguir com outra que fosse (Entrevistado 4, Commiter/RM, Colivre, [41087:43285]).

Com isso a gente termina a análise das táticas utilizadas pelos líderes para influenciar no processo de decisão e aumentar a porosidade da comunidade em relação aos desejos de seus membros desenvolvedores comuns. Um resumo analítico dessas táticas e seu componente criativo pode ser encontrado na Tabela 6 abaixo.

Táticas utilizadas pelos LÍDERES			
Tática	Breve descrição	Ocorrência na comunidade Noosfero	Componente criativo (atuação criativa dos líderes, recombinando e criando novas situações que aumentaram a porosidade na decisão sobre incorporação de código)
Encorajar os membros	Tática que consiste em encorajar os membros a continuar contribuindo e se interessar em fazer parte do grupo central de desenvolvedores	Essa foi uma tática utilizada conscientemente pelas lideranças do Noosfero. Por outro lado era limitada pelo estabelecimento de uma política técnica incompreensível para os agentes externos à Colivre	
Evitar a fragmentação	Tática que consiste em se adequar às necessidades dos membros para evitar a fragmentação da comunidade (forks definitivos)	Foi bastante utilizada na comunidade Noosfero de pelo menos três formas: com os líderes flexibilizando a data de lançamentos das versões, com os líderes assumindo o ônus de suavizar o processo de atualização quando havia mudanças radicais e finalmente com os líderes	Para essa tática encontramos o componente criativo "Tecnopolítico" que consistiu na construção de uma arquitetura de plugins no software que viabilizou um aumento no fluxo de incorporação de código principalmente nos momentos de centralização por parte da Colivre

		rejeitando contribuições que, embora boas para o software, pudessem causar traumas nos outros grupos e aumentar o risco de fragmentação	

Tabela 6: Resumo das táticas utilizadas pelos líderes que culminaram com o aumento da porosidade da comunidade em relação aos desenvolvedores comuns
Fonte: Produção própria

5. Conclusões

Essa pesquisa buscou identificar hipóteses de como a agência criativa dos membros de uma comunidade FLOSS influenciou sua história institucional, através de comportamentos e táticas que levaram à progressão de autoridade lateral e acúmulo de autoridade prática para aumento da influência na decisão sobre incorporação de código. Para isso uma análise no repositório de código da comunidade permitiu validar os principais membros para a realização das entrevistas, ou seja, aqueles que além de ter contribuído bastante com o software também estavam presentes numa diversidade de momentos destacados, identificados a partir dos próprios dados. A partir dessa definição, entrevistas semi-estruturadas foram realizadas, e sua transcrição foi codificada buscando encontrar como a agência criativa dos indivíduos contribuiu para a comunidade ter conseguido permanecer unida a despeito dos conflitos e enorme heterogeneidade entre as organizações que faziam parte. Nessa mesma análise, também buscamos identificar os comportamentos relativos à progressão de autoridade e práticas de construção institucional, bem como as táticas utilizadas pelos membros para influenciar a tomada de decisão na comunidade. Também foram buscadas duas táticas utilizadas pelos líderes que culminaram em aumento de porosidade da comunidade. Nas sessões abaixo apontaremos algumas finalidades práticas das conclusões obtidas com essa pesquisa. O diagrama abaixo oferece um mapa geral dos achados da pesquisa.

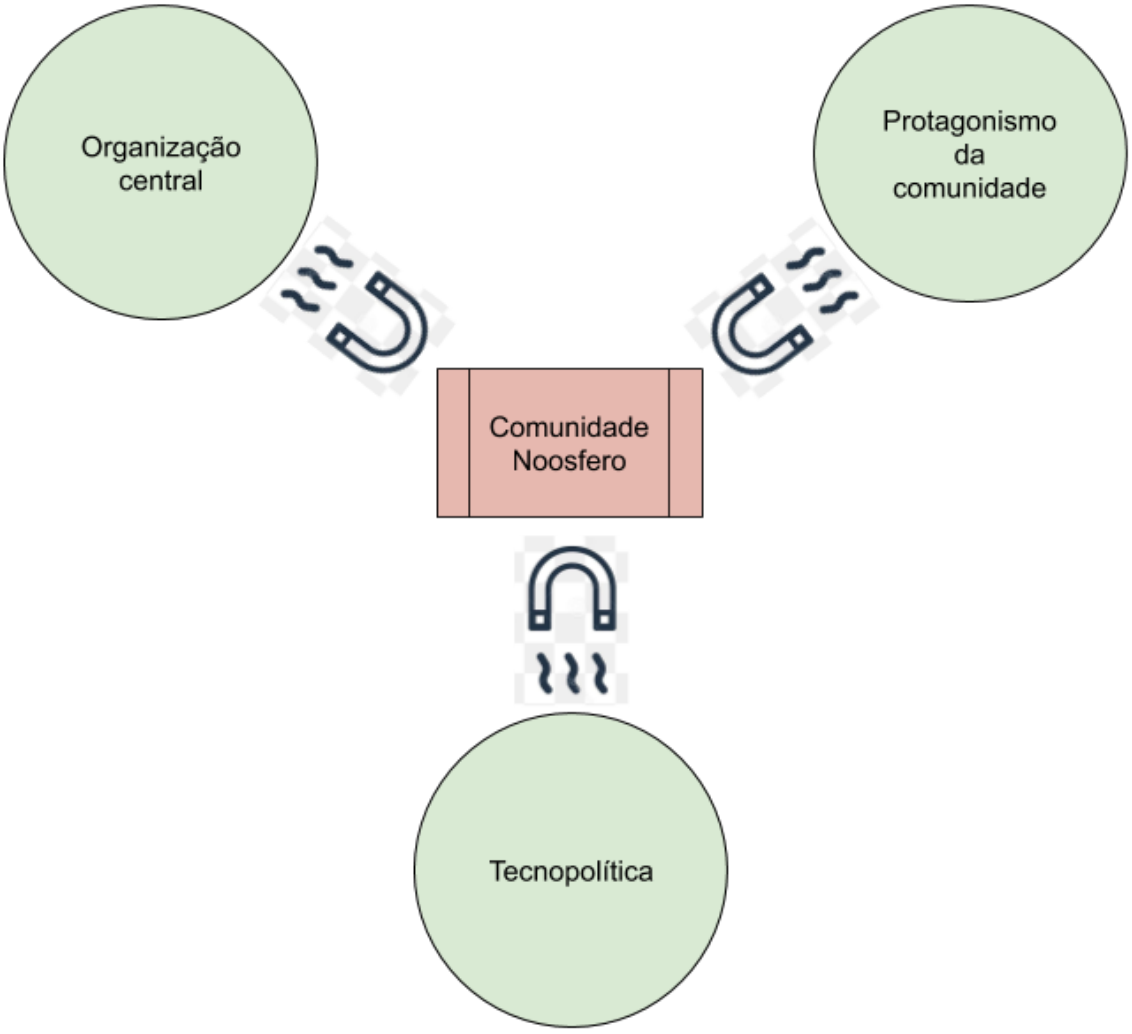

Figura 11. Fatores responsáveis pela façanha da comunidade Noosfero de ter permanecido unida

Fonte: Produção própria

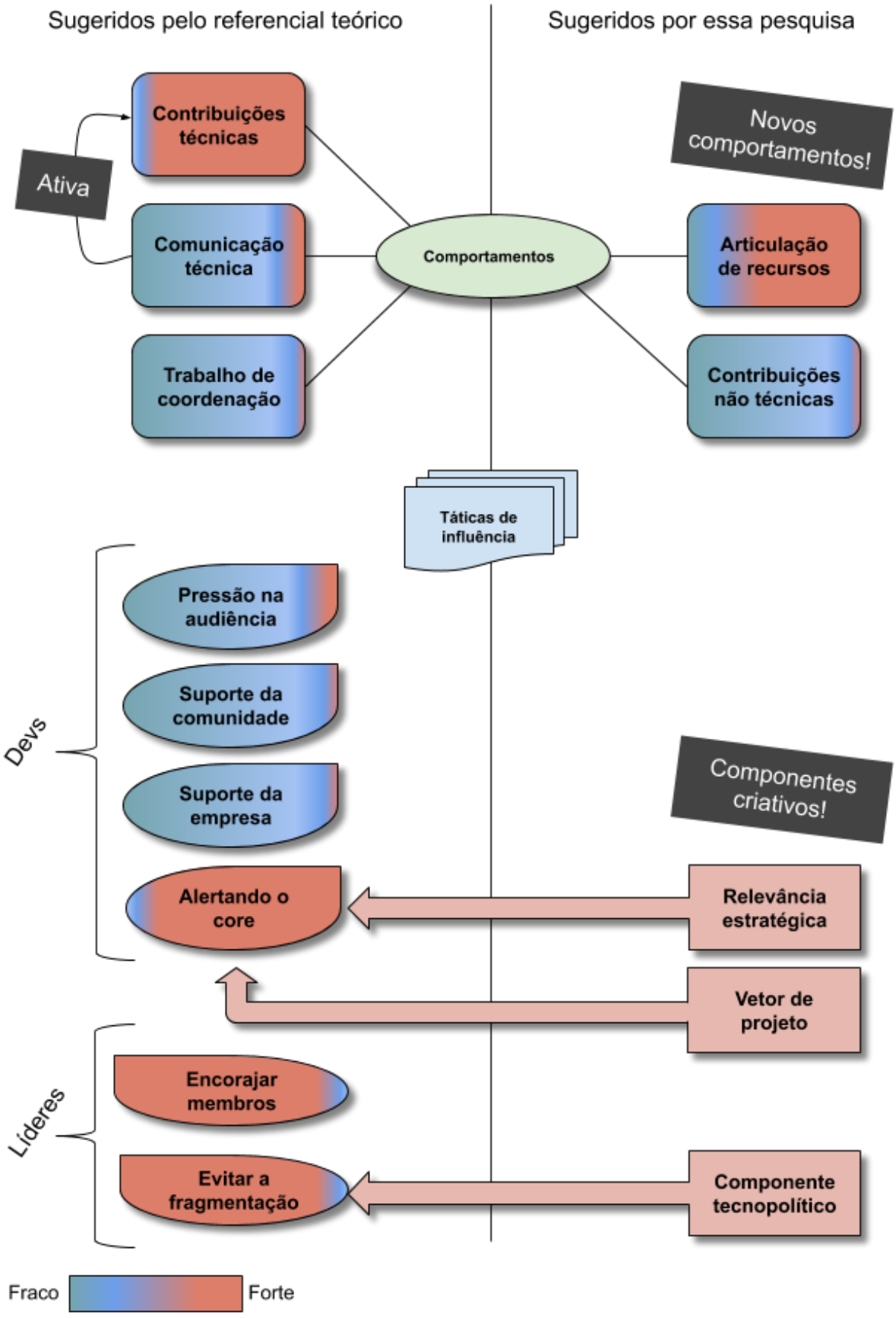

Figura 12. Comportamentos e táticas para influenciar a decisão sobre o código

5.1. Comunidades FLOSS são instituições amarradas pela agência

O exemplo do Noosfero nos mostrou que uma organização formada em torno de um Comum de software é uma instituição frágil. Parte disso se deve à uma característica intrínseca do FLOSS que pode a qualquer momento ser copiado por qualquer pessoa ou grupo e uma nova comunidade surgir. O custo para a divisão é muito baixo, já que nenhuma organização é capaz de limitar o acesso de qualquer pessoa ao Comum. Por outro lado há inúmeras vantagens em participar de uma comunidade diversa. Um software com uma base de usuários grande é um software mais conhecido, aumentando a reputação dos seus desenvolvedores. Um software com uma comunidade ativa de desenvolvedores é um software que evolui e não fica parado no tempo. Quanto mais grupos participando do seu desenvolvimento, mais rapidamente o software cresce e menos falhas ele tende a ter. Esses incentivos são percebidos pelos agentes que trabalham conscientemente para evitar a divisão da comunidade. Se por um lado o fato da comunidade Noosfero ser formada por organizações tão heterogêneas entre si tornou esse desafio mais ousado (façanha), por outro o contexto de pouca rigidez das regras (que na pesquisa comparamos com o conceito de "entrelaçamento institucional" de Abers & Keck, 2013) criou oportunidades ímpares para a ação criativa dos atores.

Dessa forma, o que vimos foram atores usando da criatividade para construir soluções de governança que fossem capaz de manter a comunidade unida. Vimos atores disputarem politicamente a entrada nas áreas mais centrais da comunidade para com isso poder pacificar os interesses de suas próprias organizações e evitar a fragmentação da comunidade. Vimos desenvolvedores comuns e líderes construírem soluções tecnopolíticas para dar vazão às necessidades das organizações e preservar o sentido de seguir fazendo parte da comunidade. Também vimos atores investirem na sua relação pessoal com os membros mais centrais da comunidade, investindo recursos de projeto nessa aproximação e potencializando a formação de lideranças próprias para ocupar esses espaços. Vimos líderes se adequando às necessidades dos outros membros para que a comunidade mantivesse o dinamismo necessário para revelar novas lideranças e seguir recebendo contribuições para o aprimoramento constante do software. Todas esses exemplos e hipóteses de formas de agência que identificamos e

sistematizamos ao longo da pesquisa demonstram que a permanência da comunidade como uma única instituição foi fruto do trabalho exaustivo e diverso de muitos atores diferentes. Se tivéssemos que apontar uma inércia inerente a comunidade, essa seria a de se fragmentar. A fragmentação da comunidade Noosfero em diversos grupos com bases de código distintas só não aconteceu por conta desse trabalho árduo e criativo executado pelos atores que fizeram parte dela.

Isso foi o que vimos na comunidade Noosfero. Vimos uma instituição dependente da agência dos atores para permanecer unida e com isso se desenvolver. A proposição que fazemos é que essa característica pode ser de alguma forma comum às instituições desse tipo (comunidade FLOSS), tanto por compartilharem essa peculiar característica do livre acesso ao Comum (o código) quanto pelos nítidos incentivos coletivos em se preservar a comunidade fortalecida e unida. Em outras palavras, a hipótese é que uma comunidade FLOSS é uma instituição que deve sua unidade à agência, ou seja, é uma instituição amarrada pela agência. Por outro lado, a história é repleta de casos de comunidades FLOSS que se dividiram já que, como na política, nem sempre as tentativas de composição são suficientes para atender todos os interesses em jogo. E é isso que torna a façanha da comunidade Noosfero ainda mais interessante.

5.2. Adotei um FLOSS, e agora?

A história do Noosfero nos permite perceber que os rumos de uma comunidade FLOSS estão sempre em disputa. Vimos como o movimento de economia solidária que, a despeito de ter sido um dos responsáveis pela criação do Noosfero, teve que passar boa parte do tempo disputando espaço para para que suas contribuições fossem consideradas pelo grupo central de desenvolvedores. Ao mesmo tempo, vimos que sem a presença da Colivre como uma organização central, é pouco plausível que o Noosfero tivesse conseguido garantir sua existência como um software único utilizado para uma grande variedade de objetivos. Mesmo nos momentos mais críticos a Colivre garantia a publicação das novas versões, sendo cuidadosa em não quebrar o uso de nenhuma outra organização da comunidade. Nunca vamos saber o que aconteceria se a Colivre tivesse abandonado esse papel, mas o fato é que nenhuma

outra organização disputou esse espaço (manifestado pela posição de RM), como vimos nas análises.

"Quem sai na chuva, é pra se molhar", já diria o ditado. Partindo do princípio de que "pensar de forma sistêmica como as práticas funcionam pode ser útil na prática" (Abers & Keck, 2013, p. 209) queremos oferecer essa pesquisa para quem está vivendo esses desafios na prática. Se sua organização adotou um FLOSS você é nosso público alvo e deve se preparar para dialogar com uma comunidade. O caso do Noosfero nos demonstra que mesmo em um FLOSS comercial a influência sobre os rumos do software está sujeita à práticas de construção institucional e comportamentos desenvolvidos no interior da instituição, ou seja, fazendo parte do coletivo da comunidade. A utilização da tática de se aproximar dos desenvolvedores centrais, seja através do "vetor de projeto" ou pela via da "relevância estratégia", requer que o agente faça parte da comunidade e passe por seu processo de socialização. Mesmo a competência de construir os atalhos tecnopolíticos para resolução de problemas de governança dependem de um conhecimento aprofundado da base de código e dos principais dilemas do coletivo, assim como da necessidade de compor com outras organizações da comunidade. Em outras palavras, ao adotar um FLOSS, os atores da organização devem estar preparados para se envolver e participar das dinâmicas da comunidade, disputando seus rumos e celebrando as vitórias coletivas. Dessa forma, a instituição que adota um FLOSS deve criar certas capacidades que lhe permita navegar nesse mar institucional, a princípio desconhecido, e ter condições de lançar mão das práticas de construção institucional, comportamentos e táticas aqui apontados, além de outras que não foram detectados pelo nosso radar. Porque no final das contas, adotar um FLOSS é também uma espécie de casamento com um ser vivo de outra espécie, que vai requerer certo trabalho de adaptação para que a organização possa usufruir do valor oferecido por aquela comunidade e ao mesmo tempo contribuir no aprimoramento do empreendimento coletivo do FLOSS.

5.3. Pontos para pesquisa futura

Essa pesquisa se propôs estudar um arranjo social de comunidade FLOSS que tem como principal ambiente de trabalho uma plataforma de codificação social. Essa característica permitiu que o pesquisador tivesse acesso a dados granulares da interação dos

desenvolvedores com o Comum de software mantido pela comunidade. A análise desses dados permitiu identificar alguns padrões que demonstraram uma grande heterogeneidade entre a atuação de cada membro nos diversos períodos ao longo do tempo. A identificação de momentos com grandes mudanças e a dispersão dessas mudanças no código trazem um indicativo para pesquisas futuras que tenham como objetivo estudar de forma específica cada um desses momentos. Por uma questão do escopo dessa pesquisa ter sido estabelecer um primeiro contato com as dinâmicas de progressão e acúmulo de autoridade de forma mais geral, não foi possível focar especificamente em cada um desses momentos. Porém essa metodologia de análise dos dados das contribuições poderá ser utilizada por pesquisas futuras que queiram entender como essas dinâmicas acontecem em momentos específicos e sua relação com os momentos destacados identificados na plataforma de codificação social.

Como pesquisa futura, as hipóteses de progressão de autoridade lateral e acúmulo de autoridade prática podem ser testadas em outras comunidades que mantenham um FLOSS orientado a serviço ("service-oriented"), tanto da mesma dimensão que a comunidade Noosfero, quanto maiores. Perceber se os comportamentos, táticas e componentes criativos identificados na comunidade Noosfero encontram paralelo em outras comunidades e se eles ajudam a explicar as dinâmicas de autoridade será importante para testar o potencial de generalização das hipóteses aqui apresentadas. Além disso, a agenda de pesquisa sobre agência criativa dos indivíduos na construção de soluções tecnopolíticas também merece um aprofundamento específico, ajudando no mapeamento desse tipo de tática e aumentando o nosso conhecimento em relação aos exemplos de quando tecnologia foi desenvolvida para superar conflitos e exercer práticas de construção institucional em instituições baseadas em plataformas digitais.

Por fim, é possível aprofundar no tema das capacidades necessárias para as organizações que adotem FLOSS e se veem obrigadas a interagir no contexto de comunidades virtuais. Conectando com um ponto que trouxemos na sessão sobre relevância do tema, a tendência mundial de adoção de FLOSS pelo Estado para dar suporte a políticas e serviços digitais à população aumenta a relevância de pesquisas que tenham como objetivo analisar as capacidades Estatais necessárias às organizações do Estado na interação com comunidades FLOSS, independente da área de política pública. Essas pesquisas poderiam beber das hipóteses aqui apresentadas aplicando-as de forma transversal na temática do uso de tecnologia da informação baseadas em FLOSS pelo Estado, oferecendo um conhecimento

mais aprofundado para que gestores possam levar a cabo a implementação dessas políticas dentro da burocracia estatal.

6. Referências

ABERS, Rebecca; KECK, Margaret E., Practical Authority: Agency and Institutional Change in Brazilian Water Politics, [s.l.]: OUP USA, 2013.

AKSULU, Altay; WADE, Michael. A Comprehensive Review and Synthesis of Open Source Research. [s.l.: s.n.], 2010.

BADASHIAN, Ali Sajedi; ESTEKI, Afsaneh; GHOLIPOUR, Ameneh; et al. Involvement, Contribution and Influence in GitHub and Stack Overflow. In: Proceedings of 24th Annual International Conference on Computer Science and Software Engineering. Riverton, NJ, USA: IBM Corp., 2014, p. 19–33. (CASCON '14). Disponível em: <http://dl.acm.org/citation.cfm?id=2735522.2735527>. Acesso em: 5 mar. 2018.

BENKLER, Yochai. The political economy of commons. Upgrade: The European Journal for the Informatics Professional, v. 4, n. 3, p. 6-9, 2003.

BENKLER, Yochai. The Wealth of Networks: How Social Production Transforms Markets and Freedom. New Haven: Yale University Press, 2006.

CASTELLS, Manuel. A Galáxia da Internet: reflexões sobre a Internet, os negócios e a sociedade. Rio de Janeiro: Jorge Zahar Ed, 2003

CASTELLS, Manuel. A Sociedade em Rede. São Paulo: Paz e Terra, 2008.

COLOMBO, Clelia, Innovación democrática y TIC, ¿hacia una democracia participativa?, IDP. Revista de Internet, Derecho y Política, n. 3, 2006.

CRIADO, J. Ignacio. Las administraciones publicas en la era del gobierno abierto. Gobernanza inteligente para un cambio de paradigma en la gestion publica. Revista de Estudios Políticos, n. 173, 2016. Disponível em: <http://www.cepc.gob.es/Publicaciones/Revistas/revistaselectronicas?IDR=3&IDN=1361&IDA=37798>. Acesso em: 25 fev. 2018.

DAHLANDER, Linus; O'MAHONY, Siobhan. Progressing to the center: Coordinating project work. Organization science, v. 22, n. 4, p. 961-979, 2011.

DABBISH, Laura; STUART, Colleen; TSAY, Jason; et al. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work. New York, NY, USA: ACM, 2012, p. 1277–1286. (CSCW '12). Disponível em: <http://doi.acm.org/10.1145/2145204.2145396>. Acesso em: 24 fev. 2018.

FORTE, Andrea; LAMPE, Cliff, Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature, American Behavioral Scientist, v. 57, n. 5, p. 535–547, 2013.

FREEMAN, Jo. The tyranny of structurelessness. Berkeley Journal of Sociology, p. 151-164, 1972.

FREITAS, Christiana Soares de; MEFFE, Corinto. Redes de produção de conhecimento tecnológico: um projeto governamental brasileiro. Estudos de Sociologia, v. 15, n. 29, 2010. Disponível em: <http://piwik.seer.fclar.unesp.br/estudos/article/view/2978>. Acesso em: 26 nov. 2017.

FREITAS, Christiana Soares de. O Software Público Brasileiro: novos modelos de cooperação econômica entre Estado e Sociedade Civil. Informação & Sociedade: Estudos, v. 22, n. 2, 2012. Disponível em: <http://periodicos.ufpb.br/ojs2/index.php/ies/article/view/12231>. Acesso em: 1 mar. 2018.

GERBAUDO, Paolo. 2016. "Social media teams as digital vanguards: the question of leadership in the management of key Facebook and Twitter accounts of Occupy Wall Street, Indignados and UK Uncut". Information, Communication & Society, published online 29 March, 1-18.

GERMANI, Leonardo Barbosa. Desafios para o desenvolvimento de serviços digitais pelo governo federal brasileiro. 2016. Disponível em: tede2.pucsp.br/tede/handle/handle/18772. Acesso em: 26 nov. 2017.

GOLD, Jon. Which countries have open-source laws on the books? Disponível em: https://www.networkworld.com/article/3114619/open-source-tools/which-countries-have-open-source-laws-on-the-books.html. Acesso em: 07/outubro/2017.

HESS, Charlotte; OSTROM, Elinor; INSTITUTO DE ALTOS ESTUDIOS NACIONALES (EQUADOR), Los bienes comunes del conocimiento, Madrid: Traficantes de Sueños : IAEN, Instituto de Altos Estudios Nacionales del Ecuador, 2016.

KALLIAMVAKOU, Eirini; DAMIAN, Daniela; SINGER, Leif; et al. The codecentric collaboration perspective: Evidence from GitHub," DCS-352-IR. [s.l.: s.n.], 2014.

KUK, George. Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List. Management Science, v. 52, n. 7, p. 1031–1042, 2006.

KRANICH, Nancy. Para contrarrestar el cercamiento, recuperar los bienes comunes del conocimiento. Los bienes comunes del conocimiento, p. 107-142, 2016.

LAVAL, Christian; DARDOT, Pierre, Común: Ensayo sobre la revolución en el siglo XXI, [s.l.]: Editorial GEDISA, 2014.

MARTÍNEZ-TORRES, M. R. A genetic search of patterns of behaviour in OSS communities. Expert Systems with Applications, v. 39, n. 18, p. 13182–13192, 2012

MARTINEZ-TORRES, Rocio; C. DIAZ-FERNANDEZ, M. Current issues and research trends on open-source software communities. [s.l.: s.n.], 2014.

MCDONALD, Nora; GOGGINS, Sean. Performance and Participation in Open Source Software on GitHub. In: CHI '13 Extended Abstracts on Human Factors in Computing Systems. New York, NY, USA: ACM, 2013, p. 139–144. (CHI EA '13). Disponível em: <http://doi.acm.org/10.1145/2468356.2468382>. Acesso em: 5 mar. 2018.

MEIRELES, Adriana Veloso. Democracia 3.0 : interação entre governo e cidadãos mediada por tecnologias digitais. 2015. Disponível em: <http://repositorio.unb.br/handle/10482/19044>. Acesso em: 26 nov. 2017.

MELUCCI, Alberto. Challenging Codes: Collective Action in the Information Age. [s.l.]: Cambridge University Press, 1996.

MENDONÇA, Ricardo Fabrino; AMARAL, Ernesto F. L., Racionalidade online: provimento de razões em discussões virtuais, Opinião Pública, v. 22, n. 2, p. 418–445, 2016.

MERGEL, Ines. Open collaboration in the public sector: The case of social coding on GitHub. Government Information Quarterly, v. 32, n. 4, p. 464–472, 2015.

NAKAKOJI, Kumiyo; YAMAMOTO, Yasuhiro; NISHINAKA, Yoshiyuki; et al. Evolution Patterns of Open-source Software Systems and Communities. In: Proceedings of the International Workshop on Principles of Software Evolution. New York, NY, USA: ACM, 2002, p. 76–85. (IWPSE '02). Disponível em: <http://doi.acm.org/10.1145/512035.512055>. Acesso em: 15 fev. 2018.

O'MALEY, Daniel. Software Público Brasileiro (SPB): The State in the Commons. In: WORKSHOP SOBRE SOFTWARE LIVRE. Anais do WSL2013. Porto Alegre-RS: SBC. [s.l.: s.n.], 2013, p. 1–10.

REYES LÓPEZ, Arturo. Analyzing GitHub as a Collaborative Software Development Platform: A Systematic Review. 2017. Disponível em: <https://dspace.library.uvic.ca//handle/1828/7953>. Acesso em: 25 fev. 2018.

ROSSI, Maria Alessandra. Decoding the Free/Open Source Software Puzzle. [s.l.: s.n.], 2006.

SALDAÑA, Johnny, The coding manual for qualitative researchers, Thousand Oaks, CA: Sage Publications Ltd, 2009.

SANTANNA, R. Sociedades democráticas precisam compartilhar seus códigos. Computer World. 16 jul. 2007. Disponível em: <https://softwarepublico.gov.br/social/spb/noticias/software-publico-em-artigo-do-secretario-rogerio-santanna>. Acesso em: 17 mar. 2018

SHIRKY, Clay. Lá vem todo mundo: o poder de organizar sem organizações. Rio de Janeiro: Zahar, 2012.

SØRENSEN, Eva; TORFING, Jacob. Enhancing Collaborative Innovation in the Public Sector. Administration & Society, v. 43, n. 8, p. 842–868, 2011.

TORAL, S. L.; MARTÍNEZ-TORRES, M. R.; BARRERO, F. J. Virtual communities as a resource for the development of OSS projects: the case of Linux ports to embedded processors. Behaviour & Information Technology, v. 28, n. 5, p. 405–419, 2009.

TSAY, Jason; DABBISH, Laura; HERBSLEB, James. Let's Talk About It: Evaluating Contributions Through Discussion in GitHub. In: Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York, NY, USA: ACM, 2014, p. 144–154. (FSE 2014). Disponível em: <http://doi.acm.org/10.1145/2635868.2635882>. Acesso em: 5 mar. 2018.

VASILESCU, Bogdan; FILKOV, Vladimir; SEREBRENIK, Alexander. Perceptions of Diversity on GitHub: A User Survey. In: Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering. Piscataway, NJ, USA: IEEE Press, 2015, p. 50–56. (CHASE '15). Disponível em: <http://dl.acm.org/citation.cfm?id=2819321.2819330>. Acesso em: 5 mar. 2018.

VAZ, J. C.. Governança Eletrônica: para onde é possível caminhar? Instituto Pólis, 2005. Disponível em:http://www.polis.org.br/uploads/745/745.pdf. Acesso em: 07/outubro/2017.

VAZ, José Carlos. Transformações tecnológicas e perspectivas para a gestão democrática das políticas culturais. Cadernos Gestão Pública e Cidadania, v. 22, n. 71, 2017.

VON BÜLOW, Marisa. 2017. “The survival of leaders and organizations in the digital age: lessons from the Chilean student movement”. Mobilization, no prelo.

WEBER, Steven. “The Success of Open Source”. Harvard University Press, 2004. 312 p.

ZHANG, Yang; WANG, Huaimin; YIN, Gang; et al. Exploring the Use of @-mention to Assist Software Development in GitHub. In: Proceedings of the 7th Asia-Pacific Symposium on Internetware. New York, NY, USA: ACM, 2015, p. 83–92. (Internetware ’15). Disponível em: <http://doi.acm.org/10.1145/2875913.2875914>. Acesso em: 5 mar. 2018.

ANEXO A - PASTAS CONSIDERADAS NA ANÁLISE DOS COMMITS

Listamos abaixo quais pastas foram utilizadas na análise e as que foram removidas, com os seus respectivos motivos.

Pastas consideradas na análise:

noosfero/util
noosfero/test
noosfero/spec
noosfero/script
noosfero/plugins
noosfero/lib
noosfero/features
noosfero/etc
noosfero/debian
noosfero/db
noosfero/cypress
noosfero/config
noosfero/bin
noosfero/baseplugins
noosfero/app

Pastas retiradas da análise:

PASTA	JUSTIFICATIVA PARA RETIRADA DA ANÁLISE
/doc	Documentação do software (bem básica), irrelevante do ponto de vista de contribuição de código
/index	Pasta temporária usada apenas no inicio do projeto, irrelevante
/log	Pasta temporária de logs, ou seja, sua alteração não é relevante para o software
/po	Pasta com os arquivos de tradução, isoladamente não refletem novas funcionalidades, sendo assim, irrelevante para nossa análise
/public	Pasta com arquivos de softwares externos, irrelevante por não fazerem parte do código do Noosfero
/vendor	Pasta com arquivos de softwares externos, irrelevante por não fazerem parte do código do Noosfero

ANEXO B - FILTRAGEM DE MOMENTOS DESTACADOS

A tabela abaixo explica como que essa filtragem foi feita. A coluna “Qtd de momentos inicialmente pinçados” exibe a quantidade inicial de momentos que foi pinçada e a coluna “Qtd de momentos selecionados para a análise” exibe a quantidade de momentos de cada tipo que foram efetivamente selecionados para a lista final. A tabela seguinte consolida a lista de 19 momentos selecionados.

Tipos de momentos, quantidade inicialmente pinçados e quantidade de selecionados para a lista final:

Fatores presentes no momento chave	Qtd de momentos inicialmente pinçados	Qtd de momentos selecionados para a análise
FATOR MUITA COISA NOVA (Fig 5): Muita mudança com saldo muito grande de adição / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita adição, pouca diferença branches x master	6	3
FATOR MUITA COISA NOVA (Fig 5): Muita mudança, muita remoção de código / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita subtração, pouca diferença branches x master	3	2
FATOR MUITA COISA NOVA (Fig 5): Muita mudança, muita remoção de código	1	1
FATOR MUITA COISA NOVA (Fig 5): Muita mudança, mas pouca diferença entre adição e subtração de código	5	2
FATOR MUITA COISA NOVA (Fig 5): Muita mudança, muita remoção de código / FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): muito commit, bem espalhado / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita subtração, pouca diferença branches x master	1	1
FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): Muito commit, bem concentrado	4	2
FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): Muito commit, bem concentrado / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita adição, pouca diferença branches x master	1	1

FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): muito commit, bem espalhado / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita adição, pouca diferença branches x master	1	1
FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): Pouco commit, muito espalhado	2	2
FATOR TENDÊNCIA DE ESPALHAMENTO (Fig. 6): Pouco commit, muito espalhado / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita adição, pouca diferença branches x master	1	1
FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita diferença entre branches e master com pouco saldo de adição ou subtração	10	3
FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita adição, pouca diferença branches x master	2	0
FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS (Fig 7): Muita subtração, pouca diferença branches x master	1	0

Lista final dos 19 momentos chave:

Semana, Data	**Fator que motivou a seleção do momento**	**Nomes dos atores que submeteram atualizações ao ramo principal nessa semana específica**
W42, Oct 15 2007	FATOR TENDÊNCIA DE DISPERSÃO: Muito commit, bem concentrado	AntonioTerceiro LeandroNunes MoisesMachado ValessioBrito
W48, Nov 26 2007	FATOR TENDÊNCIA DE DISPERSÃO: Muito commit, bem concentrado	AntonioTerceiro LeandroNunes MoisesMachado
W23, Jun 02 2008	FATOR TENDÊNCIA DE DISPERSÃO: Muito commit, bem concentrado / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita adição, pouca diferença branches x master	AntonioTerceiro AurelioAHeckert JoenioCosta MoisesMachado
W24, Jun 09 2008	FATOR MUITA COISA NOVA: Muita mudança, muita remoção de código	AntonioTerceiro JoenioCosta MoisesMachado

W42, Oct 17 2011	FATOR MUITA COISA NOVA: Muita mudança, muita remoção de código / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita subtração, pouca diferença branches x master	Braulio Bhavamitra Joenio Costa Larissa Reis Rodrigo Souto
W43, Oct 22 2012	FATOR TENDÊNCIA DE DISPERSÃO: muito commit, bem disperso / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita adição, pouca diferença branches x master	Alessandro Palmeira + João M. M. da Silva Alessandro Palmeira + Paulo Meirelles + João M. M. da Silva Braulio Bhavamitra Caio SBA Daniela Soares Feitosa João M. M. da Silva + Alessandro Palmeira Larissa Reis Paulo Meirelles + Alessandro Palmeira + João M. M. da Silva Rafael Reggiani Manzo Rodrigo Souto
W52, Dec 24 2012	FATOR MUITA COISA NOVA: Muita mudança, muita remoção de código / FATOR TENDÊNCIA DE DISPERSÃO: muito commit, bem disperso / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita subtração, pouca diferença branches x master	Caio SBA Daniela Soares Feitosa Joenio Costa Rafael Martins
W22, May 27 2013	FATOR MUITA COISA NOVA: Muita mudança, muita remoção de código / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita subtração, pouca diferença branches x master	Antonio Terceiro Aurélio A. Heckert Daniela Soares Feitosa Lucas Melo Rodrigo Souto
W26, Jun 24 2013	FATOR TENDÊNCIA DE DISPERSÃO: Pouco commit, muito disperso / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita adição, pouca diferença branches x master	Aurélio A. Heckert Daniela Soares Feitosa Leandro Nunes dos Santos vfcosta
W03, Jan 12 2015	FATOR MUITA COISA NOVA: Muita mudança com saldo muito grande de adição / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita adição, pouca diferença branches x master	Antonio Terceiro Braulio Bhavamitra Daniela Feitosa Gabriela Navarro Larissa Reis Maurilio Atila Rodrigo Souto Tallys Martins Victor Costa

W15, Apr 06 2015	FATOR MUITA COISA NOVA: Muita mudança com saldo muito grande de adição / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMO: Muita adição, pouca diferença branches x master	Aurélio A. Heckert Braulio Bhavamitra Gabriela Navarro Larissa Reis Leandro Nunes dos Santos Luciano Prestes Cavacanti Rodrigo Souto Victor Costa
W24, Jun 08 2015	FATOR MUITA COISA NOVA: Muita mudança com saldo muito grande de adição / FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita adição, pouca diferença branches x master	André Guedes Antonio Terceiro Arthur Del Esposte Aurélio A. Heckert Braulio Bhavamitra Gabriela Navarro Joenio Costa Rodrigo Souto Tallys Martins Victor Costa
W39, Sep 21 2015	FATOR MUITA COISA NOVA: Muita mudança, mas pouca diferença entre adição e subtração de código	Antonio Terceiro Arthur Del Esposte Aurélio A. Heckert Braulio Bhavamitra Daniela Feitosa Larissa Reis Marcos Ronaldo Rodrigo Souto
W43, Oct 19 2015	FATOR TENDÊNCIA DE DISPERSÃO: Pouco commit, muito disperso	Arthur Del Esposte Braulio Bhavamitra Marcos Ronaldo
W44, Oct 26 2015	FATOR MUITA COISA NOVA: Muita mudança, mas pouca diferença entre adição e subtração de código	Antonio Terceiro Daniela Feitosa Larissa Reis Michel Felipe de Oliveira Ferreira Thiago Ribeiro
W07, Feb 15 2016	FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita diferença entre branches e master com pouco saldo de adição ou subtração	Antonio Terceiro Braulio Bhavamitra Carlos Purificacao Eduardo Vital Leandro Nunes dos Santos Rafael Reggiani Manzo Rodrigo Souto Tallys Martins Victor Costa

W16, Apr 18 2016	FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita diferença entre branches e master com pouco saldo de adição ou subtração	Braulio Bhavamitra Carlos Purificacao Evandro Junior Iryna Pruitt Joenio Costa Larissa Reis Leandro Nunes dos Santos Luciano Prestes Cavalcanti Macartur Sousa Marcos Ronaldo Rafael Reggiani Manzo Rodrigo Souto Tallys Martins Victor Costa
W34, Aug 21 2017	FATOR TENDÊNCIA DE DISPERSÃO: Pouco commit, muito disperso	Gabriel Silva Leandro Nunes dos Santos Rodrigo Souto Victor Costa
W21, May 21 2018	FATOR DIFERENÇA DO PRINCIPAL COM OUTROS RAMOS: Muita diferença entre branches e master com pouco saldo de adição ou subtração	Alax Alves Gabriel Silva Jesús Parrillas Leonardo Ribeiro Soares - SUPDE/DESDR/DE507 Matheus Richard mendesiasmin Rodrigo Souto

ANEXO C - TABELA COM OS CÓDIGOS VIA DESCRIPTIVE CODING

Apresentamos abaixo uma tabela onde é possível ver a lista de códigos criadas a partir da utilização da técnica “Descriptive Coding” com uma breve explicação de seu significado. Foram códigos criados com base nos assuntos que surgiram da leitura da transcrição e que ao mesmo tempo possuíam relação com as perguntas colocadas pela pesquisa. Alguns códigos que criados na primeira rodada de codificação, foram posteriormente unidos com códigos mais relevantes, totalizando uma lista final de 21 códigos obtidos através desse método. Para fins de organização no software de coding, todos os códigos criados dessa forma iniciam com o caractere “underline”: “_”.

Nome do coding	Descrição do coding
_affordancesdaPlataforma	Conteúdo ligado às “affordances” da plataforma de codificação social que influenciam o processo de colaboração e a tomada de decisão
_ativismo	Conteúdo ligado ao exercício do ativismo no contexto da comunidade e do desenvolvimento do software
_backboneORG	Conteúdo ligado à principal organização da comunidade, a Colivre, cooperativa de software que iniciou o desenvolvimento do software e que manteve seu protagonismo durante todo o tempo de vida da comunidade
_comunicacao	Conteúdo ligado à comunicação entre os desenvolvedores e organizações da comunidade
_comunidadeORG	Conteúdo ligado à organização da comunidade
_confianca	Conteúdo ligado a relação de confiança entre os membros
_economiaSolidaria	Conteúdo ligado ao movimento de Economia Solidária na comunidade
_faxina	Conteúdo ligado ao trabalho de faxina de código na comunidade
_InfraTechparaColaboracao	Conteúdo ligado aos componentes tecnológicos que influenciam no trabalho de colaboração
_janeladeIncorporacao	Conteúdo ligado à oportunidade de incorporação de código

	pela comunidade
_machismo	Relatos sobre machismo
_meritocracia	Conteúdo onde meritocracia foi citada
_MinhaAtuacaoProfissional	Conteúdo ligado à atuação profissional dos membros
_momentoChaves	Conteúdo ligado aos momentos importantes para a comunidade
_poder	Conteúdo ligado a questões que envolvem poder na comunidade
_politicadeDesenvolvimento	Conteúdo ligado à política de incorporação de código da comunidade
_ProjetoLivreComercial	Conteúdo ligado as questões comerciais do software
_recusadeCodigo	Conteúdo ligado à recusa de incorporação de código pela comunidade
_responsabilidade	Conteúdo com menções ao aspecto da responsabilidade dos membros em relação à questões da comunidade
_rixa	Conteúdo ligado a conflitos na comunidade
_softwareLivre	Conteúdo ligado aos aspectos do software livre

Tabela. Códigos criados a partir da técnica “descriptive coding”

Fonte: Produção própria

ANEXO D - ROTEIRO BASE UTILIZADO NAS ENTREVISTAS

Chamada da entrevista: Obrigado pela disponibilidade em participar da entrevista, se trata de um estudo para uma dissertação de mestrado na área da ciência política. Meu objetivo é contribuir para o campo de estudos da ciência política com o fenômeno da colaboração aberta na contrução de softwares livres. O resultado da minha pesquisa pode ajudar no entendimento desse fenômeno tornando-o mais inteligível para gestores públicos ou demais atores do campo da política pública.

Para começar a entrevista, como nasceu o noosfero?

Me conta quando e por quê você começou a participar da construção do Noosfero

PERGUNTA PARA DEIXAR NA MANGA: Pode descrever como foi a sua trajetória na comunidade, desde o primeiro contato até o papel que desempenha hoje? Quais eram suas pretensões?

Você considera que a comunidade Noosfero tem um objetivo comum? Qual seria? Se as resposta for não, acha que algum momento já teve?

Pra você de quem é o papel de definir o objetivo comum da comunidade?
de cada membro a partir do uso que faz do software
dos desenvolvedores mais ativos
do release manager
Outro: de quem?

Resumidamente, descreva que papéis já desempenhou na comunidade. Se puder indique a partir dessa lista de papéis comuns em comunidades de software livre
Lista pra oferecer ao entrevistado:
(1) Fez/Faz parte do grupo que toma as principais decisões sobre o software
(2) Contribui/u de forma ativa (pelo menos duas vezes por mês) mas não se envolve em decisões
(3) Contribui/u esporadicamente (poucas vezes por ano) e não se envolve em decisões
(4) Posta contribuições e registra bugs
(5) Só pergunta nos fóruns e listas

Em algum momento da sua atuação na comunidade vc chegou a atuar de forma mais ativa (mais de duas vezes por mês)? sim não

Enumere até 3 contribuições técnicas que considera ter feito para a comunidade durante o período que participou/participa

Enumere até 3 contribuições não ligadas à código que você fez para a comunidade durante o mesmo período (Ex: responder perguntas sobre seu código, coordenar alguma frente de trabalho, organizar repositório, coordenar alguma atividade da comunidade)

Durante o tempo que participou/participa da comunidade chegou a desenvolver alguma atividade remunerada envolvendo noosfero? Se sim, pode falar da relação desses projetos com a comunidade?

Você é (ou foi) commiter da comunidade? Em que período?

Você já foi release manager da comunidade? Em que período?

SO EXECUTA PARA QUEM É COMMITER DA COMUNIDADE

Quais as suas principais atividades como commiter?

Quais as vantagens de ser um commiter? (ou desvantagens, pode aparecer também)

Dentre as três opções, qual mais se aproxima da maneira com que você enxerga o papel de commiter?

(i) exclusivamente técnico: avaliar a qualidade dos códigos e o que está pronto (ou não) para entrar na versão
(ii) técnico e gerencial: olhar a qualidade do código, mas também sugerir ou definir a pertinência de uma incorporação com base no que é estratégico para a comunidade
(iii) técnico e líder: olhar a qualidade do código mas também apontar uma meta comum para a comunidade, fazendo sugestões/tomando decisões com base nisso
(iv) nenhuma das alternativas atende: qual então?

Como foi o processo para você se tornar commiter da comunidade?

Qual é o procedimento quando você quer incorporar algum código no ramo principal do software? Você conversa com outros commiters? Conversa com o release manager? Você poderia me dar um exemplo?

Como commiter você recebia muitos pedidos de outros desenvolvedores para revisar e incorporar código no ramo principal do software? Como esses pedidos eram feitos? Já chegou a ser pressionado alguma vez?

Como commiter você já recusou código em definitivo de alguém? Se sim, como você agiu nesse episódio?

Como commiter, você já se sentiu responsável pelo futuro da comunidade? Já se envolveu ativamente em discussões sobre isso? Poderia me dar um exemplo?

Como commiter você já se sentiu responsável em organizar a comunidade? Poderia me dar um exemplo?

SO EXECUTA PARA QUEM FOI RELEASE MANAGER PELO MENOS UMA VEZ

Quantas vezes foi release manager e por quanto tempo?

Como era decidida a circulação de release managers?

Quais eram suas principais atividades como release manager?

Quais eram suas principais preocupações como release manager?

Dentre as três opções, qual mais se aproxima da maneira com que você enxerga o papel de release manager?
(i) exclusivamente técnico: avaliar a qualidade dos códigos e o que está pronto (ou não) para entrar na versão
(ii) técnico e gerencial: olhar a qualidade do código, mas também sugerir ou definir a pertinência de uma incorporação com base no que é estratégico para a comunidade (iii) técnico e lider: olhar a qualidade do código mas também apontar uma meta comum para a comunidade, fazendo sugestões/tomando decisões com base nisso
(iv) nenhuma das alternativas atende: qual então?

Com que frequência você discutia com os outros desenvolvedores da comunidade sobre os códigos que seriam incoporados? Fazia distinção entre commiters e/ou desenvolvedores ativos/menos ativos?

Em que medida essa discussão influenciava a decisão de incorporar alguma mudança/funcionalidade nova/correção no branch principal do software?

Como release manager você recebia muitos pedidos de outros desenvolvedores para revisar e incorporar código no ramo principal do software? Como esses pedidos eram feitos? Já chegou a ser pressionado alguma vez?

Na maior parte das vezes quem tomava efetivamente a decisão de incorporar código no branch principal do software?

Como release manager, você já se sentiu responsável pelo futuro da comunidade? Já se envolveu ativamente em discussões sobre isso? Poderia me dar um exemplo?

Como release manager, você já se sentiu responsável em organizar a comunidade? Poderia me dar um exemplo?

Consegue lembrar quais releases te marcaram mais nessa sua trajetória como release manager? Pode descrever com exemplos o que mais te marcou nelas?

EXECUTA PARA TODOS OS ENTREVISTADOS

Agora queria fazer algumas específicas sobre as relações entre os membros da comunidade. Como é o processo de aprendizado na comunidade? Os membros mais antigos estão disponíveis para responder dúvidas básicas de novos desenvolvedores? Ou há membros específicos que desempenham esse papel?

(SE A PESSOA É/FOI RELEASE MANAGER OU COMMITER RESGATAR O TEMPO EM QUE ELA NÃO ERA) Quando você submetia um código você conversava com algum desenvolvedor mais experiente ou release manager? Isso costumava ser antes ou depois de submeter o código?

Você se lembra de algum episódio onde teve que convencer os outros desenvolvedores (ou o release manager) da importância de incorporar a sua contribuição?

Pode descrever como foi esse processo de convencimento?

Você já teve um código recusado em definitivo pela comunidade? Se sim, como você agiu nesse episódio?

Você considera que a comunidade Noosfero tem líderes? Pode indicar nominalmente alguns? (tentar obter pelo menos 3 nomes)

Para você, quem tem legitimidade/pode representar a comunidade em eventos ou falar em nome da comunidade?
Qualquer usuário ativo do software
Só quem já fez alguma contribuição de código (commit)
Somente desenvolvedores muito ativos (centrais)
Apenas o release manager

Você acredita que o papel de organizar a comunidade deve ser:
de qualquer membro da comunidade de
um dos desenvolvedores mais ativos
apenas do release manager
Outro: de quem?

Consegue lembrar algum episódio quando um trabalho de organização melhorou algo na comunidade?

PERGUNTA COMPLEXA: Consegue lembrar um exemplo em que algum membro da comunidade contribuiu para reforçar ou modificar a sua motivação de participar? Consegue lembrar de algum episódio desse tipo?

SESSÃO PARA CONFIRMAR MOMENTOS CHAVE

Considerando o impacto na camada de negócio do software (grandes mudanças, uma melhoria importante etc), você consegue apontar 3 momentos importantes na história da comunidade? Pode descrever sucintamente porque esses momentos foram importantes?